

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Localização e Navegação de AGVs Industriais

Emanuel Pereira Teixeira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: António Paulo Moreira

Orientador Interno: Héber Sobreira (INESC TEC)

Co-Orientador: Miguel Pinto (CONSOVEYO)

22 de Julho de 2018

Resumo

Existem variadas soluções de localização e navegação de AGVs no mercado baseadas em diferentes paradigmas, como por exemplo a partir da identificação de marcadores instalados no chão usado pelos robôs da *Amazon Robotics*, ou então através da identificação de marcos artificiais instalados no ambiente, ou através de balizas refletoras, ou até mesmo através do seguimento de fitas instaladas no chão. Ao mesmo tempo têm sido desenvolvidas outras soluções concorrentes a essas que necessitam de ser testadas e comparadas para que se perceba a precisão e a robustez destas comparativamente com esses sistemas comerciais já existentes. Estas tecnologias já existem no estado da arte há muitos anos e estão neste momento a chegar ao mercado, é o caso da localização baseada em contornos que tem por base o cálculo da posição do robô a partir das características do meio.

Desta forma, esta dissertação tem como objetivo o teste e comparação de sistemas de localização desenvolvidos pelo INESC-TEC com o sistema comercial da NDC8 Kollmorgen já instalado no AGV cedido pela *CONSOVEYO* e com o sistema de localização do próprio laser de navegação *SICK NAV350* posteriormente instalado no mesmo AGV. Estes dois sistemas comerciais calculam a posição e orientação do AGV tendo por base em marcos artificiais, mais propriamente com refletores instalados no ambiente de navegação. Estes serão comparados com o *Extended Kalman Filter* desenvolvido no INESC-TEC que se trata de um algoritmo de fusão sensorial e tal como os anteriores necessita que sejam instalados refletores no meio de navegação e que seja feito um mapa com as coordenadas dos mesmos para que seja possível calcular a posição e a orientação do AGV.

Por outro lado, também foi feita uma análise comparativa dos algoritmos anteriormente apresentados com um algoritmo de localização baseada em contornos, o *Perfect Match* que ao contrário dos anteriores, não necessita da instalação de refletores no ambiente de navegação, necessita apenas dos dados fornecidos pelo laser *SICK NAV350* para calcular a posição e orientação do robô.

Ainda dentro das análises comparativas foi feita a comparação entre o sistema de localização de refletores do próprio laser de navegação *SICK NAV350* com o sistema desenvolvido pelo INESC-TEC "*Laser Scan to Beacons*" para o mesmo fim. Não foi feita a comparação com o sistema de localização de refletores da NDC porque não tínhamos acesso a esses dados do sistema.

No que se refere à inicialização dos algoritmos, todos necessitam de uma estimativa da sua posição inicial para que converjam corretamente. Nesse sentido, foi proposto, implementado e testado um algoritmo de localização global que tem por base o *matching* entre as distâncias dos refletores observados e as distâncias entre o refletores do mapa de refletores que calcula a posição e orientação do robô nas coordenadas relativas ao referencial global do sistema.

Abstract

There are a variety of location and navigation solutions for AGVs in the market based on different paradigms, such as from the identification of markers installed on the floor used by Amazon Robotics robots, or by identifying artificial landmarks installed in the environment, or through reflective beacons, or even through the tracking of tapes installed on the floor. At the same time, other competing solutions have been developed that need to be tested and compared so that their accuracy and robustness can be perceived comparatively with these existing commercial systems. These technologies already exist in the state of the art for many years and are currently reaching the market, is the case of the location based on contours that is based on the calculation of the position of the robot from the characteristics of the medium.

In this way, this dissertation aims to test and compare localization systems developed by INESC-TEC with the commercial system of NDC8 Kollmorgen already installed in the AGV provided by the *CONSOVEYO* and with the localization system of the navigation laser itself *SICK NAV350* later installed on the same AGV. These two commercial systems calculate the position and orientation of the AGV based on artificial landmarks, rather with reflectors installed in the navigation environment. These will be compared with the *Extended Kalman Filter* developed in INESC-TEC which is a sensory fusion algorithm and as the previous ones requires that reflectors are installed in the navigation medium and that a map with the coordinates of the so that it is possible to calculate the position and orientation of the AGV..

On the other hand, a comparative analysis of the algorithms previously presented with a contour-based localization algorithm, *Perfect Match* that unlike the previous ones, does not need the installation of reflectors in the navigation environment, needs only the data provided by the *SICK NAV350* laser to calculate the position and orientation of the robot.

Still within the comparative analyzes was made the comparison between the localization system of reflectors of the navigation laser itself *SICK NAV350* with the system developed by INESC-TEC *Laser Scan to Beacons* for the same purpose. No comparison was made with the NDC reflector locating system because we did not have access to this system data.

With regard to the initialization of the algorithms, everyone needs an estimate of their initial position in order to converge correctly. In this sense, a global localization algorithm based on the *matching* between the distances of the observed reflectors and the distances between the reflectors of the reflector map that calculates the position and orientation of the robot in the coordinates is proposed, implemented and tested. concerning the overall system framework.

Agradecimentos

Quero agradecer aos meus pais, António Augusto Pinheiro Teixeira e Maria Manuela de Sousa Pereira, pelo empenho, dedicação, esforço e sacrifício ao longo de todo o meu percurso académico e à minha família. Quero agradecer à minha namorada Marta Canelas por estar sempre do meu lado, acompanhar o meu dia-a-dia e me apoiar em todos os meus sonhos. Quero agradecer ao Gonçalo Silva, ao Pedro Guedes e ao Pedro Henrique, que durante a faculdade formamos o "*Quarteto Fantástico*" e juntos superamos grande partes das etapas e desafios que a faculdade nos propôs. Quero agradecer também ao Pedro Pontes e ao Ricardo Urjais por serem duas das melhores pessoas que conheci até hoje e por estarem sempre ao meu lado com toda a sinceridade e amizade. E não esquecendo os restantes amigos, Filipa Magalhães, João Silva e Patrícia Campos. Quero agradecer também aos meus orientadores, António Paulo Moreira, por acreditar em mim e estar sempre pronto para ajudar para que possa alcançar os melhores resultados possíveis, ao Héber Sobreira, por estar sempre pronto a ajudar e ser incansável comigo, por último ao Miguel Pinto da Consoveyo por todo o apoio que deu durante esta dissertação.

Emanuel Pereira Teixeira

“If things are not failing, you are not innovating enough”

Elon Musk

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação e Objetivos	1
1.3	Contribuições Inovadoras	2
1.4	Estrutura da Dissertação	3
2	Revisão Bibliográfica	5
2.1	Introdução	5
2.2	Soluções Existentes	5
2.3	Localização baseada na detecção de contornos tendo por base o algoritmo <i>Perfect Match</i>	7
2.4	Localização baseada em refletores	9
2.4.1	Laser Scan to Beacons	11
3	Hardware e Software utilizados	15
3.1	AGV Consoveyo	15
3.1.1	Sistema NDC	16
3.2	SICK NAV350	17
3.2.1	Características Técnicas	18
3.2.2	Modos de Operação	18
3.2.3	SOPAS Engineering Tool	20
3.2.4	Integração de sistemas	20
3.3	Advantech ARK-3440	21
3.4	Refletores	22
3.4.1	Mapa de refletores	23
3.5	<i>Robot Operating System (ROS)</i>	23
3.5.1	Arquitetura do sistema	24
3.5.2	RVIZ	25
3.6	Matlab	26
4	Algoritmo de Localização Global para sistema de localização com refletores indistinguíveis	27
4.1	Apresentação do problema	27
4.2	Solução Proposta	27
4.3	Algoritmo de <i>Matching</i> sem <i>Outliers</i>	30
4.3.1	Função match	33
4.4	Algoritmo de <i>Matching</i> com <i>Outliers</i>	36

5	Testes e resultados	41
5.1	Ambiente de teste	43
5.1.1	Trajectoria de testes	43
5.2	Comparação da pose calculada pelo NDC com a do NAV350	45
5.3	NDC vs EKF com detecção de refletores NAV350	48
5.4	NAV350 vs EKF com detecção de refletores NAV350	51
5.5	Scan to beacons vs Detecção de Beacons NAV350	55
5.6	NDC vs EKF com detecção de refletores Laser Scan to Beacons	56
5.7	NAV350 vs EKF com detecção de refletores Laser Scan to Beacons	59
5.8	Conclusões	62
5.9	Comparações ente NDC, <i>SICK NAV350</i> e <i>Extended Kalman Filter</i> com o <i>Perfect Match</i>	64
5.9.1	<i>Perfect Match</i> vs NDC	65
5.9.2	<i>Perfect Match</i> vs NAV350	68
5.9.3	<i>Perfect Match</i> vs EKF	71
5.9.4	Conclusões	73
5.10	Algoritmo de Localização Global	75
5.11	Ground Thruth	82
6	Conclusão	85
7	Trabalho futuro	87
	Referências	89

Lista de Figuras

2.1	Arquitectura da Solução Proposta (Sobreira, 2017)	8
2.2	Representação do referencial relativo ($R_x R_y$) do robô na pose $X_v = [x_v y_v \theta_v]^T$ relativamente ao referencial do mundo $W_x W_y$. O círculo amarelo representa um refletor (elemento do mapa de refletores M_B) na posição fixa $[x_{B,i} y_{B,i}]$. As linhas tracejadas representam um conjunto de medidas obtidas com um laser scanner. $numL$ corresponde à última medida, com as coordenadas polares $Z_{L,numL} = [r_{L,numL} \phi_{L,numL}]$. Medidas com alta refletividade e baixa refletividade são diferenciadas na imagem através das cores laranja e vermelha respetivamente. (Sobreira, 2017)	10
2.3	Arquitectura da solução proposta (Sobreira, 2017)	11
2.4	A azul estão representados os resultados experimentais do modelo da relação geométrica entre o número de feixes incidentes num refletor em função da distância do laser ao centro do refletor e a vermelho está representada a resposta do modelo (Sobreira, 2017)	12
2.5	Relação geométrica entre a distância R_i a um refletor B_i e o número de feixes do laser scanner (Sobreira, 2017)	12
2.6	Filtro de outliers (Sobreira, 2017)	13
3.1	À esquerda a vista da parte lateral do AGV com o respetivo comprimento e altura e à direita a vista da parte traseira com as respetivas dimensões mínima e máxima de largura	15
3.2	LiDAR 2D SICK NAV350	17
3.3	Exemplo da deteção de um refletor no referencial local do laser	19
3.4	SOPAS Engineering tool	20
3.5	Poste de navegação combinada com suporte metálico. Limitação de visibilidade identificada à direita	21
3.6	Computador industrial Advantech ARK-3440F-U5A2E	22
3.7	Representação em CAD do refletor cilíndrico utilizado	23
3.8	Mapa do local de testes e posição dos refletores	23
3.9	Logótipo ROS Kinetic	24
3.10	Arquitectura do sistema	25
3.11	RVIZ	25
4.1	Mapa de refletores representado no referencial global com três distâncias entre refletores identificadas, D1, D2 e D3. As distâncias observadas entre refletores é invariante em relação à pose do robô e ao referencial na qual estão a ser observadas.	28
4.2	Refletores observados pelo robô no referencial do mesmo com três distâncias entre refletores identificadas, D4, D5 e D6. As distâncias observadas entre refletores é invariante em relação à pose do robô e ao referencial na qual estão a ser observadas.	29

4.3	Exemplo ilustrativo da ambiguidade que acontece quando se usa apenas uma distância para fazer <i>matching</i>	32
5.1	Mapa de ocupação do local de testes	43
5.2	Mapa do espaço feito a partir do <i>package</i> ROS Hector Mapping com a trajetória de testes que o AGV efetuou	44
5.3	Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo NAV350 ao longo da trajetória	45
5.4	Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo NAV350 ao longo da trajetória, onde é evidenciada a curva 2	46
5.5	Diferença de posição em metros entre posição dada pelo NDC e NAV350 ao longo do teste	46
5.6	Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo NAV350 ao longo da trajetória	47
5.7	Diferença de orientação em graus entre orientação dada pelo NDC e NAV350 ao longo do teste	47
5.8	Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória	49
5.9	Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, onde é evidenciada a curva 2	49
5.10	Diferença absoluta de posição em metros entre posição dada pelo NDC e EKF ao longo do teste	50
5.11	Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória	50
5.12	Diferença de orientação em graus entre orientação dada pelo NDC e EKF ao longo do teste	51
5.13	Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória	52
5.14	Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a curva 2 evidenciada	53
5.15	Diferença de posição em metros entre posição dada pelo NAV350 e EKF ao longo do teste	53
5.16	Na figura à esquerda está representada a orientação, em graus, dada pelo NAV350 ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória	54
5.17	Diferença de orientação em graus entre orientação dada pelo NAV350 e EKF ao longo do teste	54
5.18	Número de possíveis refletores detetados pelos dois sistemas ao longo da trajetória pelo nó <i>Laser Scan to Beacons</i> e pelo próprio laser <i>SICK NAV350</i>	56
5.19	Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	57

5.20	Diferença de posição em metros entre posição dada pelo NDC e EKF ao longo do teste, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	57
5.21	Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i> . .	58
5.22	Diferença de orientação em graus entre orientação dada pelo NDC e EKF ao longo do teste, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	58
5.23	Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	59
5.24	Diferença de posição em metros entre posição dada pelo NAV350 e EKF ao longo do teste, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	60
5.25	Na figura à esquerda está representada a orientação, em graus, dada pelo NAV350 ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i> . .	60
5.26	Diferença de orientação em graus entre orientação dada pelo NAV350 e EKF ao longo do teste, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	61
5.27	Mapa do espaço feito a partir do <i>package</i> ROS Hector Mapping com a trajetória de testes que o AGV efetuou nas comparações entre o <i>Perfect Match</i> e os restantes algoritmos	64
5.28	A azul está representada a posição dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a posição do sistema de localização por refletores da NDC durante a execução da trajetória	65
5.29	Diferença absoluta entre a posição dada pelo sistema da NDC e o algoritmo de localização com base em contornos <i>Perfect Match</i>	66
5.30	A azul está representada a orientação dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a orientação do sistema de localização por refletores da NDC durante a execução da trajetória	66
5.31	Diferença de orientação entre a posição dada pelo sistema da NDC e o algoritmo de localização com base em contornos <i>Perfect Match</i>	67
5.32	A azul está representada a posição dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a posição do sistema de localização por refletores do laser <i>SICK NAV350</i> durante a execução da trajetória	68
5.33	Diferença absoluta entre a posição dada pelo próprio laser <i>SICK NAV350</i> e o algoritmo de localização com base em contornos <i>Perfect Match</i>	69
5.34	A azul está representada a orientação dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a orientação do sistema de localização por refletores do laser <i>SICK NAV350</i> durante a execução da trajetória	69
5.35	Diferença de orientação entre a orientação dada pelo laser <i>SICK NAV350</i> e o algoritmo de localização com base em contornos <i>Perfect Match</i>	70
5.36	A azul está representada a posição dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a posição do algoritmo <i>Extended Kalman Filter</i> , durante a execução da trajetória	71
5.37	Diferença absoluta entre a posição dada pelo algoritmo <i>Extended Kalman Filter</i> e o algoritmo de localização com base em contornos <i>Perfect Match</i>	72
5.38	A azul está representada a orientação dada pelo algoritmo de localização com base em contornos <i>Perfect Match</i> e a vermelho a orientação do algoritmo <i>Extended Kalman Filter</i> durante a execução da trajetória	72

5.39	Diferença de orientação entre do algoritmo <i>Extended Kalman Filter</i> e o algoritmo de localização com base em contornos <i>Perfect Match</i>	73
5.40	Mapa de refletores usado no algoritmo de localização global	75
5.41	Refletores detetados pelo laser <i>SICK NAV350</i>	76
5.42	Resultado do algoritmo de localização global	76
5.43	Conjunto de refletores detetados pelo laser <i>SICK NAV350</i> com <i>outliers</i>	77
5.44	Resultado do algoritmo de localização global	78
5.45	Conjunto de refletores detetados pelo laser <i>SICK NAV350</i>	79
5.46	Resultado do algoritmo de localização global	79
5.47	Conjunto de refletores detetados pelo laser <i>SICK NAV350</i> , com <i>outliers</i> adicionados	80
5.48	Resultado do algoritmo de localização global	81
5.49	Montagem do teste para comparação entre o deslocamento registado pelo laser de medição com o deslocamento registado pelos sistemas de localização da NDC, <i>SICK NAV350</i> , <i>Extended Kalman Filter</i> e <i>Perfect Match</i>	82
7.1	Máquina de estados para integração do algoritmo de localização global com o <i>Extended Kalman Filter</i>	88

Lista de Tabelas

3.1	Especificações técnicas LiDAR 2D SICK NAV350	18
3.2	Especificações técnicas Advantech ARK-3440F-U5A2E	22
4.1	Matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.	31
4.2	Matriz $Zdist$ com as distâncias entre cada um dos refletores do mapa.	33
4.3	Exemplo de matriz $Zdist$ com as distâncias entre cada um dos refletores observado.	34
4.4	Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.	34
4.5	Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.	35
4.6	Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.	35
4.7	Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.	36
4.8	Exemplo de matriz M com as posições observados de cada possível refletor no referencial do <i>laser</i>	38
4.9	Exemplo de matriz W com as posições observados de cada possível refletor no referencial do <i>laser</i> e os pesos associados a cada uma delas.	38
5.1	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e NAV350	48
5.2	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF	52
5.3	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF	55
5.4	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	59
5.5	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NAV350 e EKF, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	61
5.6	Tabela com os valores calculados a partir dos dados das três comparações	62
5.7	Comparação entre o <i>Extended Kalman Filter</i> com o uso de detecção de refletores do SICK NAV350, e com o uso de detecção de refletores com <i>Laser Scan to Beacons</i> (LSbT) e os restantes algoritmos	63
5.8	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o <i>Perfect Match</i> e o NDC	67
5.9	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e <i>SICK NAV350</i>	70
5.10	Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF, com a posição dos refletores dada pelo nó <i>Scan to Beacons</i>	73
5.11	Comparação entre o <i>Perfect Match</i> com o <i>Extended Kalman Filter</i> , <i>SICK NAV350</i> e o sistema da NDC	74

5.12	Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com três refletores detetados sem <i>outliers</i>	77
5.13	Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com três refletores detetados com <i>outliers</i>	78
5.14	Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com sete refletores detetados sem <i>outliers</i>	80
5.15	Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com sete refletores detetados com <i>outliers</i>	81
5.16	Deslocamento registrado pelo laser de medição com o deslocamento registrado pelos sistemas de localização da NDC, <i>SICK NAV350</i> , <i>Extended Kalman Filter</i> e <i>Perfect Match</i>	83
5.17	Erro entre o deslocamento registrado pelo laser de medição com os deslocamentos registrados pelos sistemas de localização da NDC, <i>SICK NAV350</i> , <i>Extended Kalman Filter</i> e <i>Perfect Match</i>	83

Abreviaturas e Símbolos

AGV	Automatic Guided Vehicles
EKF	Extended Kalman Filter
PM	Perfect Match
ICP	Iterative Closest Point
LStB	Laser Scan to Beacons

Capítulo 1

Introdução

1.1 Contexto

AGVs, são Veículos Autonomamente Guiados capazes de desempenhar funções como rebocamento, carregamento ou empilhamento de forma autónoma. Este tipo de tecnologia é uma solução viável, atrativa e diferenciadora no que diz respeito à movimentação interna e armazenamento na indústria, pois permite diminuir a mão de obra em tarefas monótonas e repetitivas que em pouco ou nada acrescentam valor a um processo. A localização e navegação destes veículos nos dias de hoje é um grande desafio, pois existe um grande dinamismo natural num ambiente fabril e além disso, as empresas procuram que as suas fábricas sejam flexíveis e capazes se adaptarem às necessidades que a produção de novos produtos exige. Neste contexto o robô terá de se conseguir localizar e navegar em ambientes dinâmicos e em constante mudança de forma precisa, robusta e segura.

1.2 Motivação e Objetivos

Ao longo das últimas décadas foram surgindo várias soluções de localização baseadas em diferentes algoritmos e diferentes tipos de sensores. Apesar disso, a quantidade de soluções que se mostram capazes de solucionar o problema, continua a ser reduzido e aplicado a ambientes muito controlados. Tal deve-se ao facto da falta de robustez e fiabilidade dos algoritmos desenvolvidos, e ao custo das tecnologias para a utilização dos mesmos ([Reinke and Beinschob, 2013](#)). Soluções como o seguimento de faixas ([Schulze and Wullner, 2006](#)), são bastante simples e têm um custo relativamente baixo, no entanto, são pouco flexíveis para ambientes industriais bastante dinâmicos, pois a mudança da trajetória por onde o AGV se desloca, usando esta tecnologia, implica a mudança da faixa, fixa no chão, que muitas vezes acarreta custos elevados. Uma solução mais flexível é por exemplo a localização por triangulação ([Schulze et al., 2008](#)). A navegação por refletores é considerada a tecnologia de localização mais precisa, robusta e flexível e a mais utilizada em AGVs meio industrial. No entanto, esta tecnologia apresenta um custo maior que as anteriores devido à necessidade de um *laser* de navegação e a instalação de refletores no meio de navegação,

onde o robô se pode movimentar (Schulze et al., 2008) (Sobreira, 2017). Estes sistemas, para além de fechados, dependentes de equipamento externo ao veículo, são geralmente caros e ainda um pouco limitados em se adaptarem a situações imprevistas. O INESC TEC tem trabalho desenvolvido em sistemas de localização e navegação inovadores que além de minimizarem ou mesmo dispensarem os refletores são sistemas mais abertos e passíveis de melhoramentos, alterações e adaptações a diferentes situações.

Pretende-se nesta dissertação comparar o sistema de localização de um AGV industrial cedido pela CONSOVEYO, onde foi instalado um segundo laser de navegação, o *SICK NAV350*, e um computador industrial *Advantech ARK-3440*, com outros três sistemas de localização (localização por meio do próprio laser de navegação *SICK NAV350*, *Extended Kalman Filter* e o *Perfect Match*).

Em suma, serão comparados quatro sistemas de localização, três com base em refletores, são eles o sistema da NDC8 Kollmorgen, o algoritmo de localização do próprio laser *SICK NAV350* e o *Extended Kalman Filter* e um com base em contornos, o *Perfect Match*. Destes quatro sistemas, dois deles são comerciais, que é o caso do sistema da NDC8 Kollmorgen e do algoritmo de localização do próprio laser *SICK NAV350* e dois foram desenvolvidos no INESC-TEC, que é o caso do *Extended Kalman Filter* e o *Perfect Match*. Dentro disso, foram comparados os métodos de localização de refletores do sistema comercial do laser *SICK NAV350*, com o desenvolvido no INESC-TEC, o "*Laser Scan to Beacons*". Por fim foi apresentado um algoritmo de localização global para o cálculo da posição e orientação do robô no mundo.

1.3 Contribuições Inovadoras

Através da análise dos vários algoritmos de localização será possível perceber em primeiro lugar as diferenças entre os três algoritmos de localização através de refletores, NDC8 Kollmorgen, localização do laser *SICK NAV350* (ambos comerciais) e o *Extended Kalman Filter* (INESC-TEC). Em segundo será possível também perceber o comportamento de um algoritmo de localização baseada em contornos, o *Perfect Match*, ainda muito pouco utilizado comercialmente, em relação aos algoritmos anteriores já largamente utilizados comercialmente.

Em momentos como o início de operação de um AGV ou em momentos em que o robô deixa de se localizar corretamente é necessária uma estimativa da posição atual do robô para que este possa operar devidamente a partir daí. Por outras palavras, é necessário um mecanismo que calcule a posição atual do robô e a forneça aos algoritmos de localização. É proposto aqui um algoritmo de determinação da posição global do AGV, tendo por base (Ronzoni et al., 2011b). Este algoritmo é aplicável a robôs que utilizem um laser rotativo para a deteção de refletores. Refletor é uma superfície de forma cilíndrica ou retangular com um nível de refletividade alta, instalado no ambiente de navegação. Pretende-se que um refletor tenha um índice de reflexão superior ao dos restantes objetos do ambiente de navegação para que seja facilmente detetado por um laser de navegação. O algoritmo de localização global tem como objetivo determinar a correspondência entre os refletores detetados no referencial do laser com os refletores do mapa de refletores, define-se refletores detetados como o conjunto de posições relativas em relação ao referencial do robô pelo laser do

mesmo e mapa de refletores como o conjunto de posições absolutas em que os refletores previamente mapeados foram instalados. Esta correspondência é feita a partir do *matching* das distâncias entre os refletores observados, com as distâncias pré calculadas do mapa de refletores. Depois de sabermos a correspondência entre os refletores do mapa de refletores e os refletores observados, é possível calcular a transformação homogênea entre as coordenadas de ambos e calcular a posição atual do robô.

1.4 Estrutura da Dissertação

Esta dissertação é composta por sete capítulos. No capítulo 2 é feita a revisão bibliográfica dos principais algoritmos de localização utilizados em robôs móveis, onde é dada principal relevância aos algoritmos de localização por refletores e algoritmos de localização por contornos. No capítulo 3 foi descrito o *hardware* e o *software* utilizados durante a elaboração da dissertação. No capítulo 4 é proposto um algoritmo de localização global para um sistema de localização com refletores indistinguíveis entre si. No capítulo 5 são descritos os testes efetuados com todos os algoritmos testados, ou seja, a localização por parte do sistema da NDC8 Kollmorgen, a localização do próprio laser de navegação *SICK NAV350*, o *Extended Kalman Filter* e o *Perfect Match*, é testado também o algoritmo de localização global proposto e apresentados os resultados obtidos. Em último lugar são apresentadas as conclusões da dissertação no capítulo 6 e o possível trabalho futuro no capítulo 7.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

O estado da arte dos sistemas de localização é vasto e tem vindo a ser estudado há várias décadas. Existem bastantes propostas de sistemas de localização, cada uma com características diferentes tendo em conta o sensor e a classe de algoritmo utilizado. No entanto parece que as soluções comerciais existentes não dispõem de um sistema de localização com baixo custo, alta flexibilidade, alta precisão e alta robustez, simultaneamente. Tendo em conta este objetivo, ou seja, diminuir os custos, aumentar a flexibilidade, aumentar a precisão e aumentar robustez, o INESC-TEC tem trabalhado na criação e no melhoramento de algoritmos a fim de os tornar capazes de satisfazerem os exigentes requisitos de robustez e fiabilidade que por exemplo o ambiente industrial impõem.

No presente capítulo pretende-se fazer uma revisão bibliográfica dos sistemas de localização mais utilizados em AGVs industriais, dando principal foco na localização baseada em refletores e também na localização baseada em contornos. Será também feita uma revisão sobre o problema da localização global.

2.2 Soluções Existentes

Como já foi referido, o número de soluções já desenvolvidas para a localização de AGVs industriais é vasta, no entanto e de acordo com a sua aplicação, nem todas satisfazem por completo os requisitos de robustez, fiabilidade e flexibilidade que são propostos se as condições do local onde operam ou a tarefa para que foram concebidos for alterada. Uma solução robusta, simples mas com pouca flexibilidade é o caso da localização baseada em filoguiado. Como refere ([Lima et al., 2010](#)) este método é bastante utilizado pela sua simplicidade, precisão e robustez. No entanto, os únicos trajetos que o robô poderá efetuar estão restritos à introdução de um fio elétrico no chão. Trata-se de um método pouco flexível a mudanças de layout, pois se for necessário a alteração da trajetória que os robôs podem efetuar, significa a substituição do fio elétrico introduzido

no solo. Isto poderá acarretar custos e tempos de interrupção altos. No mesmo estilo de localização, temos soluções baseadas em faixas. Diferem dos sistemas filoguiados no que diz respeito à substituição do fio elétrico introduzido no chão por fitas magnéticas ou coloridas colocadas sobre o chão. Trata-se de uma solução mais flexível e barata do que os sistemas filoguiados pois, tanto a instalação, como a alteração de percursos se resume à remoção e à recolocação das fitas, processo esse que é mais rápido, barato e simples que no caso anterior. No entanto, como as fitas estão à superfície, estas podem degradar-se o que pode tornar inviável o processo de localização (Lima et al., 2010).

Existem algumas soluções que recorrem sinais *wireless* (Han et al., 2013), ou até à tecnologia RFID (*Radio-Frequency IDentification*) como é o trabalho de (Hekimian-Williams et al., 2010). No trabalho de (Zhou and Shi, 2008) é feita uma revisão das várias soluções que utilizam a tecnologia RFID. No entanto, este tipo de tecnologia pode ser inviável uma vez que é bastante sensível a atenuações e reflexões devido ao ruído eletromagnético e abundância de estruturas metálicas presentes em ambiente industrial (Sobreira, 2017) (Esteves, 2005).

Técnicas que utilizam balizas ativas emissoras de sinais infravermelhos para localização foram propostas nos trabalhos de (Pierlot and Droogenbroeck, 2014), assim como um algoritmo para triangulação (Pierlot and Droogenbroeck, 2014). Contudo este tipo de solução pode tornar-se bastante dispendiosa pois a instalação e a manutenção de balizas ativas acarreta custos elevados. Outra desvantagem é o facto de tais sinais serem afetados pela luz ambiente, no entanto, no trabalho de (Wang et al., 2017) é proposto um sistema de localização que utiliza infravermelhos, capaz de ter o factor da interferência da luz ambiente em conta e conseguir melhores resultados.

Duas das técnicas de localização mais utilizadas são a localização baseada em contornos e a localização baseada em refletores. Tanto numa solução como noutra (Thrun et al., 2005) contribui com grande parte dos algoritmos e estratégias que viabilizaram estas duas técnicas. No que se refere à localização baseada em contornos (Thrun et al., 2005) apresenta não só um filtro de partículas melhorado, que se intitula "*Augmented Monte Carlo Localization*" e uma estratégia de localização global. Outras técnicas de localização global são apresentadas em (Zhang et al., 2012), (Thrun et al., 2000). Neste tipo de abordagem é essencial a utilização de algoritmos de "*Map Matching*". Um dos algoritmos "*Map Matching*" muito utilizado, por exemplo por muitas equipas que participam em campeonatos de futebol robótico, trata-se do algoritmo "*Perfect Match*" desenvolvido por (Lauer et al., 2006). É de realçar que se trata de um algoritmo relativamente leve e que permite a sua utilização com uma frequência alta em comparação com um algoritmo da mesma linha chamado "*Iterative Closest Point*", numa comparação feita entre estes dois algoritmos em (Sobreira et al., 2016). O algoritmo "*Iterative Closest Point*" foi apresentado por (Besl and McKay, 1992) e é largamente utilizado em *scan-matching*. Em (Sobreira, 2017) é apresentado um algoritmo baseado na deteção de contornos que tem por base o algoritmo "*Perfect Match*" onde foram introduzidas algumas melhorias que permitiram melhorar o mesmo em termos de robustez e precisão, a fim de cumprirem com as exigências desses parâmetros em ambientes industriais. Trata-se de um método de localização menos dispendioso que a o da localização baseada em refletores, apresentada a seguir, uma vez que não necessita da instalação de refletores na área de

navegação do robô. Apesar de existirem várias estratégias de localização global, para auxiliarem estes algoritmos em momentos de inicialização ou quando deixam de determinar corretamente a pose do robô, essas estratégias não são muito viáveis, uma vez que implicam a circulação livre do robô para aquisição de novas medidas em trajetórias que não são as pré-definidas para a circulação do mesmo em segurança.

No que diz respeito à localização baseada em refletores (Thrun et al., 2005) propõem um algoritmo baseado num filtro de Kalman estendido chamado “*EKF Localization with Unknown Correspondences*”, aplicável à localização baseada em refletores onde os mesmos são indistinguíveis entre si. Trata-se de um método de localização bastante robusto e fiável, no entanto acarreta custos mais elevados devido ao tipo de sensor utilizado e a instalação de refletores externos ao robô. Para tentar solucionar o problema da localização global e da ocorrência de falsos positivos neste tipo de localização (Ronzoni et al., 2011a) propõe um algoritmo para tentar solucionar estes dois problemas tendo em conta as distâncias dos refletores observados com as distâncias entre os mesmos num mapa do sistema. Ainda referente à localização baseada em refletores é de realçar o trabalho desenvolvido por em (Sobreira, 2017). Neste trabalho, foi desenvolvido um sistema de localização baseada em refletores utilizando um laser de segurança. Foi feita a combinação entre um algoritmo de fusão sensorial e um método de filtragem de falsos positivos. Desta forma foi possível obter um algoritmo ainda mais robusto e com menor necessidade de informação sensorial e de menor qualidade.

2.3 Localização baseada na detecção de contornos tendo por base o algoritmo *Perfect Match*

Neste ponto é apresentado o algoritmo de localização baseado em contornos, tendo por base o algoritmo de *Matching Perfect Match*, com as alterações apresentadas em (Sobreira, 2017), tendo em vista o aumento da sua robustez e fiabilidade e a sua aplicação em ambientes exigentes como é o caso de ambientes industriais. Por alto, podemos dizer que o algoritmo proposto é a combinação de um algoritmo de fusão sensorial (*Extended Kalman Filter*) e um algoritmo de *matching*.

Como foi referido antes, o algoritmo PM, apresentado por (Lauer et al., 2006), trata-se de um algoritmo de *Matching* com um peso computacional relativamente leve comparativamente a outros algoritmos da mesma linha como é o caso do ICP. Em traços gerais este algoritmo divide-se em três etapas:

- Cálculo dos gradientes;
- Rotina de otimização baseada no *Resilient Back-Propagation*;
- Cálculo das covariâncias usando a segunda derivada.

É utilizada uma grelha de ocupação que descreve o meio envolvente ao robô da qual são calculadas três matrizes: mapa de distâncias, onde em cada célula é indicada a distância à célula

ocupada mais próxima e, dois gradientes calculados a partir das derivadas em x e em y do mapa de distâncias. Estas três matrizes são calculadas apenas uma única vez no início do programa. A arquitetura do sistema implementado em (Sobreira, 2017) é apresentada na figura seguinte:

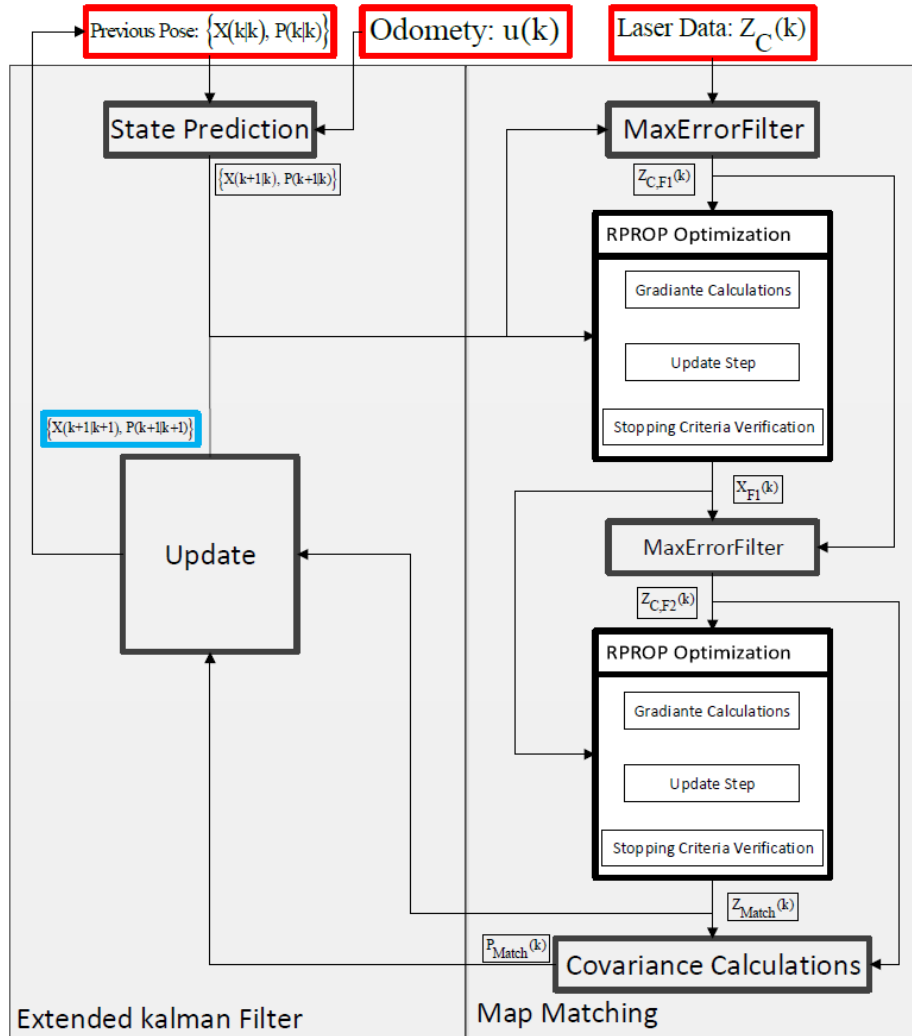


Figura 2.1: Arquitectura da Solução Proposta (Sobreira, 2017)

As entradas do sistema estão representadas a vermelho, Pose anterior, Odometria e Dados do sensor enquanto que a saída do sistema está representada a azul, ou seja, a pose do robô calculada pelo algoritmo. É possível realçar dois grandes blocos verticais, *Extended Kalman Filter* à esquerda e *Map Matching* à direita. É possível ver em mais detalhe os algoritmos no trabalho de (Sobreira, 2017). É de realçar os módulos *MaxErrorFilter* e *RPROP Optimization*. No caso do *MaxErrorFilter*, tem como objetivo de pré-processar os dados dos sensores de forma a remover falsos positivos, enquanto que *RPROP Optimization* fica encarregue de solucionar o problema de otimização definido pela função de custo (2.1).

Inicialmente neste algoritmo é utilizada a seguinte função de custo:

$$E = \sum_{i=1}^{PCount} Ei, com Ei = \frac{L_c^2}{L_c^2 + d_i^2} \quad (2.1)$$

Onde *PCount* é o número de medidas adquiridas pelos sensores, *Lc* é um parâmetro limitador da contribuição de uma dada medida com erro e *di* trata-se do valor do valor da célula no mapa de distâncias que corresponde à posição da medida *i* no referencial absoluto.

Na segunda etapa do algoritmo é corrida a otimização baseada no *Resilient Back-Propagation* onde é obtida a pose atual do robô que minimiza a função de custo (2.1).

2.3.0.1 Alterações introduzidas no algoritmo PM

No trabalho de (Sobreira, 2017) foram introduzidas algumas alterações no algoritmo PM a fim de cumprir com os valores de precisão que ambientes bastante dinâmicos exigem. Essas alterações foram:

- Melhoramento da fusão sensorial;
- Definição de um critério de convergência com um número de iterações dinâmico;

Foram também feitos testes à alteração da função de custo e uma análise da influência de falsos positivos e com o intuito de melhorar a robustez do mesmo, o problema da localização global foi tido em conta e por isso foi implementado um algoritmo de detecção de perdido e um algoritmo de localização global.

2.4 Localização baseada em refletores

Neste ponto irei apresentar os tópicos principais da localização baseada em refletores assim como a solução proposta em (Sobreira, 2017) baseada num algoritmo de fusão sensorial *Extended Kalman Filter* combinada com os módulos *Reflector Detector* e *Association/Outlier Filter*, responsáveis por pré-processar os dados usados no algoritmo de fusão sensorial e a associação de refletores com a filtragem de falsos positivos. Além disso, foi implementado um componente de supervisão, intitulado *Uncertainty Supervisor* para avaliar a incerteza da pose estimada.

Este tipo de técnica de localização pertence à família de técnicas de localização baseadas na detecção de balizas artificiais. O termo "balizas" é utilizado uma vez que a sua detecção não é feita pontualmente mas sim através de um conjunto de medidas. Estes marcos dizem-se artificiais uma vez que são colocados no ambiente de navegação do robô e portanto são externos a ele. Para que o robô estime a sua posição é necessário detetar várias destas balizas, no mínimo três, por isso, para que o robô em qualquer instância possa determinar corretamente a sua pose. Como é referido em (Sobreira, 2017), o objetivo deste método é determinar a pose $X_v = [x_v \ y_v \ \theta_v]^T$ no referencial global externo ao robô $W_x W_y$ em que $R_x R_y$ é o referencial local, móvel em relação ao referencial global, do robô 2.2.

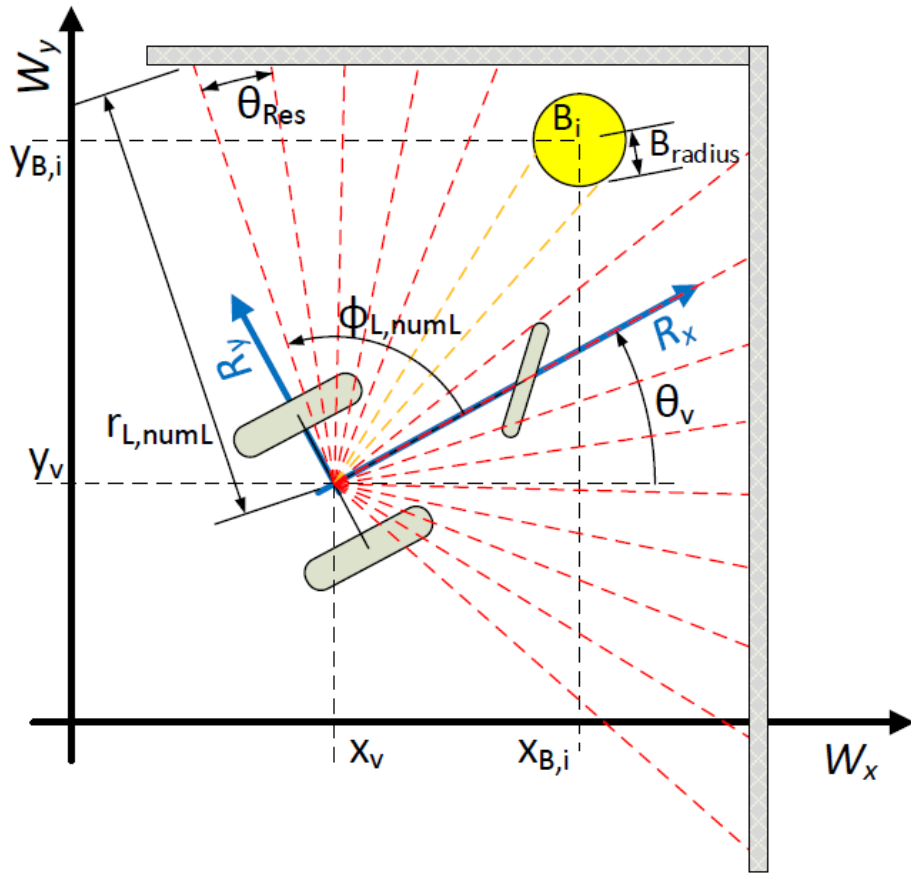


Figura 2.2: Representação do referencial relativo ($R_x R_y$) do robô na pose $X_v = [x_v, y_v, \theta_v]^T$ relativamente ao referencial do mundo $W_x W_y$. O círculo amarelo representa um refletor (elemento do mapa de refletores M_B) na posição fixa $[x_{B,i}, y_{B,i}]$. As linhas tracejadas representam um conjunto de medidas obtidas com um laser scanner. $numL$ corresponde à última medida, com as coordenadas polares $Z_{L,numL} = [r_{L,numL}, \phi_{L,numL}]$. Medidas com alta refletividade e baixa refletividade são diferenciadas na imagem através das cores laranja e vermelha respetivamente. (Sobreira, 2017)

Nesta solução é utilizado um *laser scanner* em que as medidas obtidas num dado momento são caracterizadas pelas suas coordenadas polares, no referencial do robô $R_x R_y$ como também por uma variável booleana que nos dá informação acerca da refletividade do objetivo detetado em cada uma das medidas num dado instante. Como se pode ver em 2.2, as linhas a amarelo representam medidas com alta refletividade e portanto com o parâmetro de refletividade com valor lógico igual a verdadeiro e a vermelho medidas com baixa refletividade e portanto com o valor do parâmetro de refletividade igual a falso. É possível ver também a amarelo na figura 2.2 um refletor B_i com raio B_{radius} . A variável θ_{Res} é a resolução característica do laser utilizado.

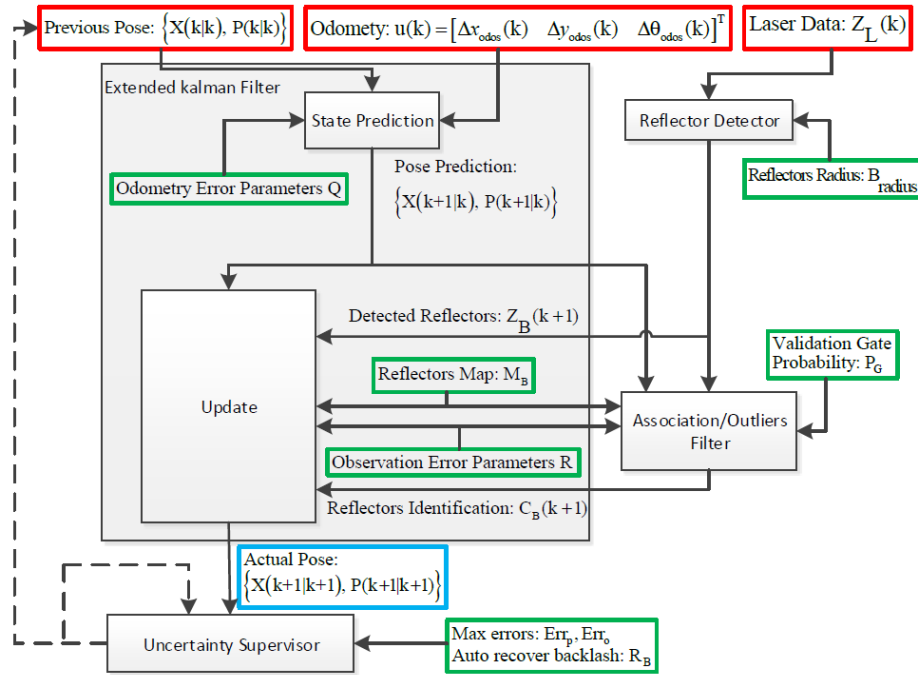


Figura 2.3: Arquitetura da solução proposta (Sobreira, 2017)

Em 2.3 é possível observar a arquitetura do sistema de localização baseada em refletores implementada em (Sobreira, 2017). Como entradas do sistema temos a pose anterior estimada, os valores da odometria e os dados do sensor, como saída do sistema temos a azul a pose atual estimada pelo sistema, por último a verde estão representados os parâmetros do sistema. É possível identificar um grande bloco a cinzento responsável pela fusão sensorial, o *Extended Kalman Filter* mas é de salientar a importância de três outros blocos adicionados neste trabalho que aumentam a robustez e a fiabilidade dos dados obtidos nesta solução em relação a dados de soluções deste tipo previamente existentes, que , são eles o *Reflector Detector*, *Association/Outliers Filter* e *Uncertainty Supervisor*.

2.4.1 Laser Scan to Beacons

O sistema proposto em (Sobreira, 2017) e utilizado neste trabalho para comparação com o sistema comercial de detecção de refletores do SICK NAV350, baseia-se no conhecimento da resolução do laser, no raio do refletor, na distância do laser ao centro do refletor e no número de feixes incidentes neste. O número de feixes incidentes num refletor depende da distância do laser ao refletor. Essa relação geométrica é dada pela equação 2.2.

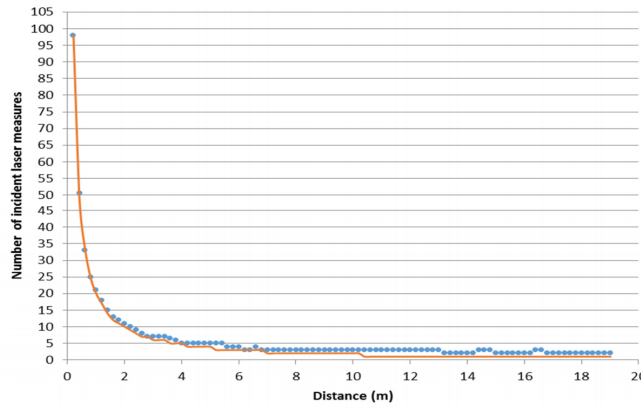


Figura 2.4: A azul estão representados os resultados experimentais do modelo da relação geométrica entre o número de feixes incidentes num refletor em função da distância do laser ao centro do refletor e a vermelho está representada a resposta do modelo (Sobreira, 2017)

Com estas variáveis, foi proposto o seguinte modelo da equação 2.2.

$$M_{num} = \text{floor} \left(\frac{2 \arcsin \left(\frac{B_{radius}}{r} \right)}{\theta_{Res}} \right) \quad (2.2)$$

Neste modelo, B_{radius} corresponde ao raio do refletor, θ_{Res} à resolução do laser e r a distância entre o laser e o centro do refletor. Como se pode verificar na figura 2.5.

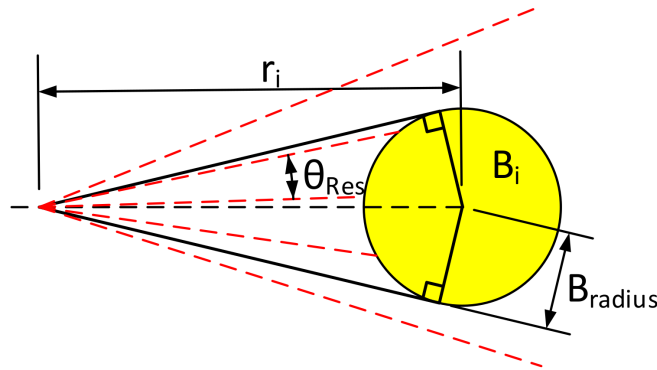


Figura 2.5: Relação geométrica entre a distância R_i a um refletor B_i e o número de feixes do laser scanner (Sobreira, 2017)

Com isto é possível ignorar os objetos que não tenham as dimensões desejadas. Além disto foi proposto um filtro para possíveis *outliers* que possam surgir. Este filtro é composto principalmente por duas funções. Uma primeira que identifica os refletores detetados e uma segunda que filtra *outliers* que possam não ter sido filtrados inicialmente, como pode ser visto no algoritmo 2.6.

Inputs : $Z_B(k), \hat{Z}_B(k), S(k)$	
1:	for all detected beacons $z_{B,i}$ do
	Association:
1.1	$C_{B,i}(k) = \underset{j}{\operatorname{argmax}} \det(2\pi S_j(k))^*$
:	
1.2	$\exp\left\{-\frac{1}{2} \left(Z_{B,i}(k) - \hat{Z}_{B,i}(k)\right)^T [S_j(k)]^{-1} \left(Z_{B,i}(k) - \hat{Z}_{B,i}(k)\right)\right\}$
:	Association filter:
1.3	if $\left(Z_{B,i}(k) - \hat{Z}_{j-C_{B,i}(k)}(k)\right)^T \left(S_{j-C_{B,i}(k)}(k)\right)^{-1} \left(Z_{B,i}(k) - \hat{Z}_{j-C_{B,i}(k)}(k)\right) > \chi_{df}^2$ then
:	
1.3	$C_{B,i}(k) = \text{INVALID_BEACON_ID}$
:	endif
	endfor
Outputs : $C_B(k)$	

Figura 2.6: Filtro de outliers (Sobreira, 2017)

Durante a associação cada refletor detetado $Z_{B,i}$ é associado a um elemento do mapa de refletores $M_{B,i}$ que lhe maximiza a probabilidade. Para cada elemento j do mapa M_B é calculada a função de distribuição de probabilidades $Z_{B,i}$. O observação i é associada a j quando esta maximiza a sua probabilidade. No segundo passo, χ_{df}^2 é uma constante que dependo do *threshold* mínimo de validação. Este valor está relacionado com a função inversa do χ probabilidade de forma quadrática para dois graus de liberdade.

Como resultado deste algoritmo, C_B consiste numa matriz onde $C_{B,i}$ contém o índice no mapa de refletores de $M_{B,i}$ associado à detecção $Z_{B,i}$.

Capítulo 3

Hardware e Software utilizados

Neste capítulo será feita a apresentação e descrição do *hardware* e *software* utilizados na elaboração desta tese. O AGV utilizado e equipado com os sistemas da *NDC* foi cedido pela *CONSOVEYO* e trata-se do elemento que será comparado com os sistemas desenvolvidos pelo *INESC-TEC*. Para que essa comparação fosse possível, foi necessária a integração desses sistemas no AGV da *CONSOVEYO*. Essa integração também será descrita neste capítulo.

3.1 AGV Consoveyo

Este veículo trata-se de um AGV empilhador com tração em triciclo e tem cerca de 3,408 metros de comprimento desde a ponta dos garfos, que corresponde à sua traseira, até ao painel de instrumentos, que corresponde à sua frente. Tem também cerca de 2,365 metros de altura, desde o solo até ao topo do laser rotativo, colocado no topo do AGV. A sua largura é variável de acordo com a abertura dos garfos, que pode ir desde 1,420 metros até 2,180 metros.

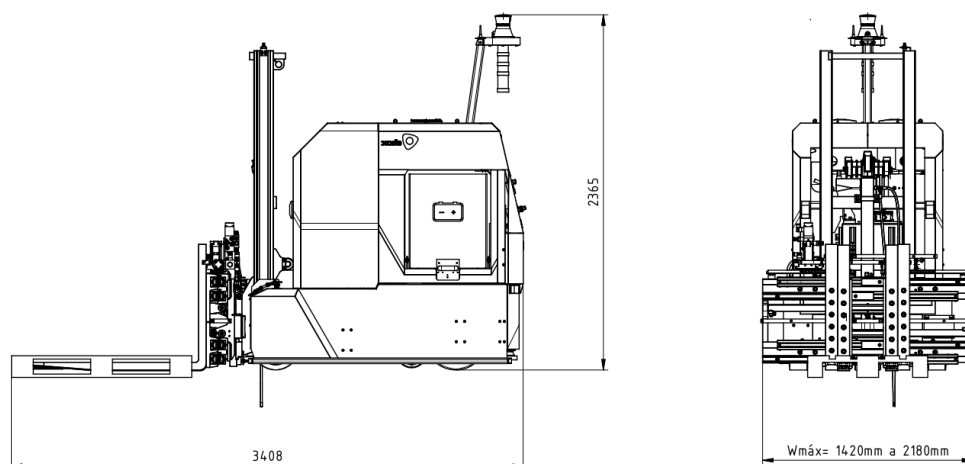


Figura 3.1: À esquerda a vista da parte lateral do AGV com o respetivo comprimento e altura e à direita a vista da parte traseira com as respetivas dimensões mínima e máxima de largura

3.1.1 Sistema NDC

A utilização do sistema da NDC presente no AGV foi limitada. Desse sistema apenas conseguimos receber informação acerca da posição atual do robô, o ângulo da roda frontal, a velocidade de tração e a distância percorrida com base nos impulsos dados pelo *encoder*, entre ciclos. No que diz respeito a comandos que era possível enviar para o AGV apenas conseguimos controlar a velocidade de tração e o ângulo da roda frontal.

Trama de dados enviada pelo sistema da NDC:

- NDC8.Position.X - Posição X;
- NDC8.Position.Y - Posição Y;
- NDC8.Position.THETA - Ângulo θ ;
- NDC8.Position.Valid - 1 se a posição enviada é válida, 0 se inválida;
- NDC8.CAN_Enc_30.SteerEncAngle - Ângulo da roda frontal;
- NDC8.MI08_20012_EFA_1_5.Actualspeed - Velocidade da tração;
- NDC8.MI08_20012_EFA_1_5.DriveEncDist - O que o AGV andou entre ciclos;
- CycleTime - Tempo de ciclo do AGV.

Trama de dados enviada para o sistema da NDC:

- Setspeed - Velocidade da tração;
- Setangle - Ângulo pretendido para a roda frontal.

Todas as medidas de posição e velocidade são em mm ou mm/s.

Para que fosse possível fazer uma comparação mais exaustiva e detalhada entre sistemas seria útil termos o acesso a todo o sistema presente no AGV. Uma vez que isso não foi possível, as comparações com o sistema do AGV tiveram como forma de comparação a posição *X*, *Y* e *Theta* dadas pelo sistema da NDC.

Além disto foi introduzido inicialmente, por parte da *Consoveyo*, o mapa de refletores do local de testes, que apenas pode ser alterado pelos mesmos. Outra das coisas que foi configurada inicialmente e que não pode ser alterada depois foi a trajetória que o robô executa. Seria interessante, para obter resultados ainda mais precisos ter havido a possibilidade de alteração destes dois pontos. Isto porque podem existir erros nas posições dos refletores configuradas em relação às posições reais dos mesmos e um conjunto de testes para várias trajetórias permitiria executar uma coletânea de testes mais exaustivos e resultados mais precisos.

3.2 SICK NAV350

O *SICK NAV350* é um sensor LiDAR 2D de longo alcance e com um ângulo de visão de 360°. Este laser será o utilizado para obter medições não só dos contornos do ambiente como também da posição dos refletores detetados pelo mesmo. Estes dados serão utilizados nos algoritmos de localização para determinar a posição do AGV e compará-la com a posição fornecida pelo sistema da NDC.

As medições deste laser estão representadas em coordenadas polares bidimensionais. Mas, para além da distância e ângulo de uma dada medição, também é fornecida a intensidade do feixe refletido.

Por si só, o laser de navegação SICK NAV350 tem a funcionalidade de se localizar globalmente. Por isso, além de fornecer as medições para serem utilizadas em algoritmos de localização, este laser já fornecerá uma pose para comparação tanto com o sistema da NDC, como com os algoritmos de localização implementados pelo INESC-TEC, como é o caso do *Extended Kalman Filter*.

Além da funcionalidade de localização este laser de navegação também disponibiliza a possibilidade de localização e identificação de refletores.



Figura 3.2: LiDAR 2D SICK NAV350

3.2.1 Características Técnicas

Aplicação	Indoor
Ângulo de Visão	360°
Frequência do scan	8 Hz \pm 5%
Resolução Angular	0.25°
Gama de operação	min 0.5 m, max 250 m (min 0.5 m, max 70 m em detecção de refletores)
Precisão de posição	\pm 4 mm
Erro sistemático	\pm 10 mm (RDI) \pm 15 mm (Contornos)
Erro estatístico	\pm 15 mm (Contornos)

Tabela 3.1: Especificações técnicas LiDAR 2D SICK NAV350

3.2.2 Modos de Operação

O SICK NAV350 disponibiliza dois modos de operação, o modo de **detecção de refletores** e o modo de **navegação**. Com a possibilidade de detetar refletores, este sistema é capaz de disponibilizar também a posição relativa dos refletores, a posição absoluta do robô, ou uma combinação das duas. Tanto no modo de detecção de refletores como no modo de navegação o sistema disponibiliza as medições dos contornos do ambiente envolvente.

No modo de **detecção de refletores** o NAV350 cria uma imagem dos refletores que estão à sua volta em cada varrimento de 360°, ou seja, a cada 125 ms. Num ambiente onde o número de refletores seja muito alto, o NAV350 seleciona os 40 refletores com intensidade mais alta de um máximo de 250 refletores.

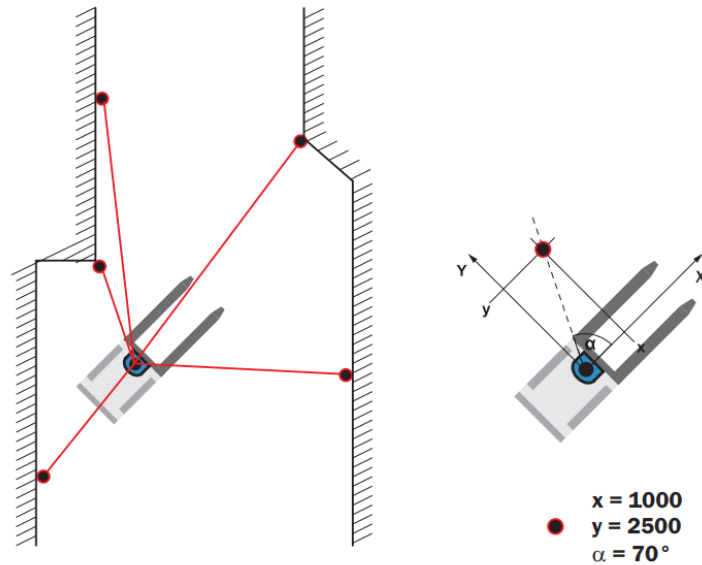


Figura 3.3: Exemplo da detecção de um refletor no referencial local do laser

Este modo permite que o computador do AGV acesse diretamente em tempo real às coordenadas dos refletores medidas pelo NAV350. Com estes dados e com um algoritmo específico é possível calcular a posição do robô. Este modo de funcionamento será especialmente útil para testar o algoritmo de localização global implementado.

Já no modo de **navegação** o laser é capaz de calcular a posição do AGV no referencial global, durante o funcionamento do AGV. O cálculo da posição do robô é baseada na detecção de refletores que se encontram em posições fixas e pré definidas. Uma vez que a determinação da posição é obtida por triangulação, a detecção de três refletores de cada vez é suficiente para determinar a posição do robô. Para que este modo possa ser colocado em funcionamento é necessário configurar duas coisas, são elas:

- Um sistema de coordenadas global;
- Um mapa com os refletores, definido em coordenadas cartesianas no referencial global.

Este modo de operação é composto por três estados, são eles:

- Posicionamento inicial;
- Posicionamento contínuo;
- Posicionamento virtual.

No estado de posicionamento inicial é despoletado assim que o laser é ligado no modo de navegação. Aqui, é calculada e identificada a posição dos refletores através da comparação dos

padrões do conjunto de refletores medidos com o mapa de refletores. Para que a posição seja corretamente calculada é necessário que o robô esteja parado e seja medidos pelo menos três refletores.

Após a determinação da posição no estado de posicionamento inicial, o NAV350 passa automaticamente para o estado de posicionamento contínuo em que o AGV pode estar a operar normalmente. Neste modo o laser disponibiliza a posição do robô com uma frequência de 8 Hz, ou seja, a cada 125 ms.

Por ultimo, caso o estado de posicionamento inicial falhe, é estimada uma posição tendo por base o modelo de velocidade e movimento. Este estado está ativo durante uma distância percorrida máxima de 3 metros. Após o robô percorrer essa distância o estado é parado.

3.2.3 SOPAS Engineering Tool

O SOPAS Engineering Tool é um software de configuração desenvolvido pela SICK AG e tem como principal função a configuração dos sensores da marca. Além da configuração dos sensores, no caso no NAV350, é possível por exemplo verificar as medidas atuais do laser, embora que não seja em tempo real, ver também os refletores detetados a cada instante. Isto tem o principal interesse em confirmar se as configurações e o funcionamento são o esperado.

Foi através deste software que o mapa de refletores e o referencial global, necessários para o funcionamento em modo de navegação, foram carregados para o laser.

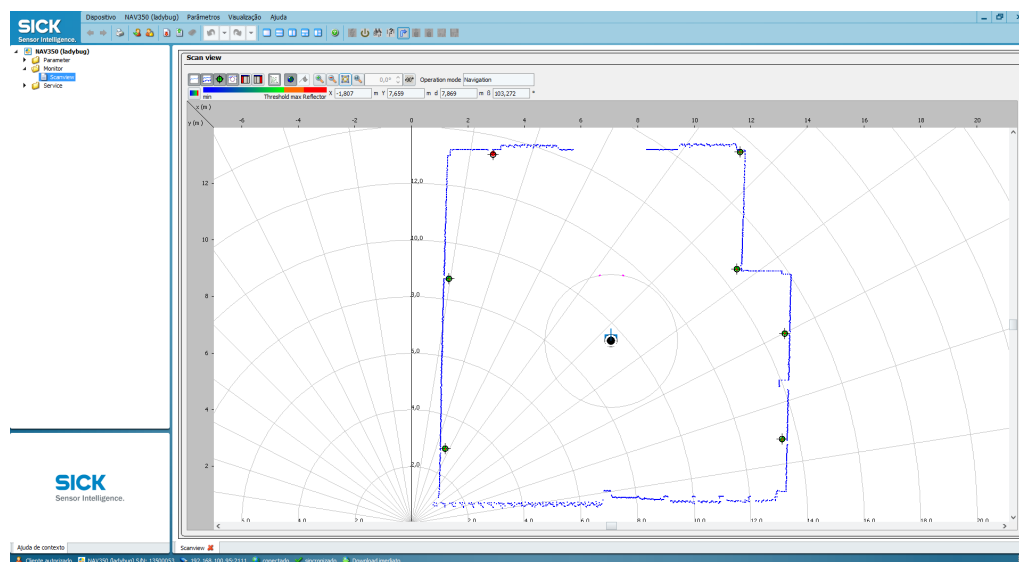


Figura 3.4: SOPAS Engineering tool

3.2.4 Integração de sistemas

Para podermos ter uma comparação, em simultâneo, o mais precisa possível entre o sistema já instalado no AGV e os sistemas do INESC-TEC foi necessário integrar o laser (SICK NAV350)

numa posição próxima do laser já instalado no robô. Para isso foi colocado um poste de navegação combinada onde na plataforma inferior foi colocado o laser SICK NAV350 e na parte superior o laser KOLLMORGEN LS5, como pode ser visto na figura 3.5.

No entanto, esta solução tem um problema. O suporte metálico vertical que une a base onde está instalado o laser inferior e a base onde está instalado o laser superior retira ângulo de visão ao laser SICK NAV350. Embora essa peça de metal seja fina, o facto de estar bastante próxima do laser, faz com que retire cerca de 25° no campo de visão do mesmo. O suporte em questão está identificado na figura 3.5.

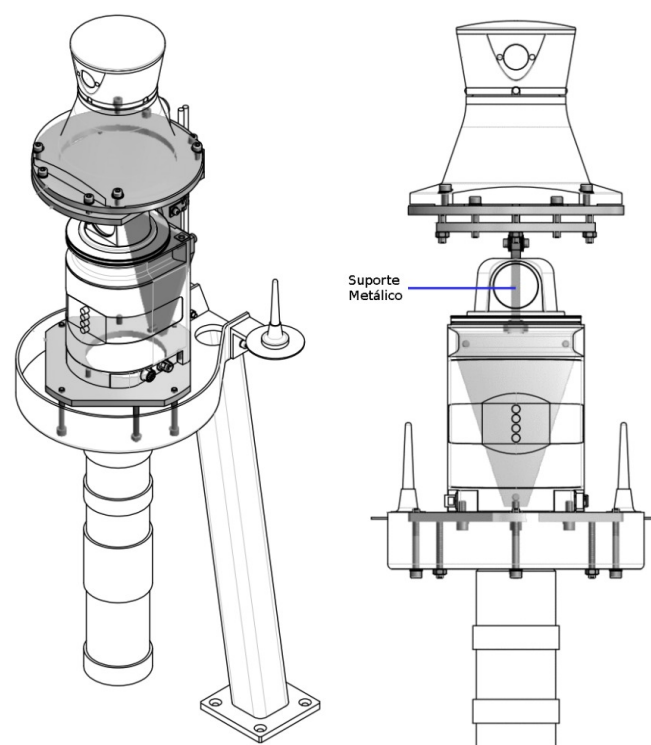


Figura 3.5: Poste de navegação combinada com suporte metálico. Limitação de visibilidade identificada à direita

3.3 Advantech ARK-3440

O computador utilizado para processar os dados recebidos pelo laser NAV350 foi um computador industrial da Advantech, modelo ARK-3440F-U5A2E, representado na figura 3.6. Este computador foi colocado na parte superior do AGV.



Figura 3.6: Computador industrial Advantech ARK-3440F-U5A2E

CPU	Intel Core i7 610E (Dual Core) 2.53 GHz
Chipset	Intel QM57
Memória	8 GB DDR3
Conectores	3xDB9, DVI-I, HDMI, 2xRJ45 Ethernet, 6xUSB, Audio (in/out/Mic), DC input (terminal block), eSATA
Input Voltage DC	9 ... 34 V
Consumo	53 W

Tabela 3.2: Especificações técnicas Advantech ARK-3440F-U5A2E

3.4 Refletores

Os refletores utilizados são refletores de forma cilíndrica, com 5 cm de diâmetro e 75 cm de altura. Estes foram fixados à parede a uma altura de 2.30 m do chão. Na figura 3.7 está representado em CAD um desses refletores utilizados.



Figura 3.7: Representação em CAD do refletor cilíndrico utilizado

3.4.1 Mapa de refletores

O local onde o robô realizou os testes e onde os refletores foram instalados trata-se de um pavilhão com cerca de 120 metros quadrados. Na figura 3.9 é possível ver um esquema do espaço e assinalado com pontos pretos o local onde os refletores foram instalados.

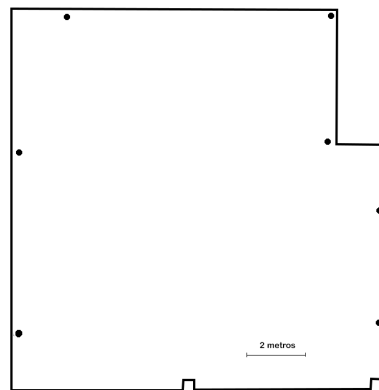


Figura 3.8: Mapa do local de testes e posição dos refletores

3.5 Robot Operating System (ROS)

Em primeiro lugar será feita uma apresentação da *framework* de desenvolvimento utilizada durante esta tese, o *Robot Operating System* (ROS) muito utilizada para desenvolvimento de *software* de aplicações de robótica. Existem várias versões desta *framework* no entanto a utilizada durante esta tese foi a versão *Kinetic*.



Figura 3.9: Logótipo ROS Kinetic

Esta *framework*, como foi dito anteriormente, é muito utilizada nos últimos tempos para desenvolvimento de *software* para aplicações de robótica e muito devido a isso possui uma grande coleção de bibliotecas desenvolvida pela vasta comunidade que usa esta ferramenta. Isto permite a reutilização de código e o desenvolvimento mais rápido de aplicações.

Numa visão de alto nível, uma sistema desenvolvido em ROS é um conjunto de programas, os "Nós", escritos maioritariamente em linguagem *C++* ou *Python*, que podem comunicar entre si através de "Tópicos" que são publicados e subscritos, e consultar informações de outros nós através de "Serviços".

A escolha sobre esta *framework* deveu-se ao facto de o *software* até então desenvolvido e a ser testado nesta tese foi desenvolvido nesta ferramenta e também pelas vantagens atrás referidas.

3.5.1 Arquitetura do sistema

O robô e o sistema da NDC presente nele funciona como uma caixa negra que fornece alguns dados, como a pose, executa uma dada trajetória pré definida e além disso pode ser controlado o ângulo da roda frontal e a velocidade do robô. Para ser possível esta comunicação foi criado um nó para fazer a interface entre o sistema da NDC e o ROS que corre no PC do sistema do INESC-TEC. Além desse nó, foi desenvolvido um nó que funciona como driver para configuração e comunicação com o laser SICK NAV350.

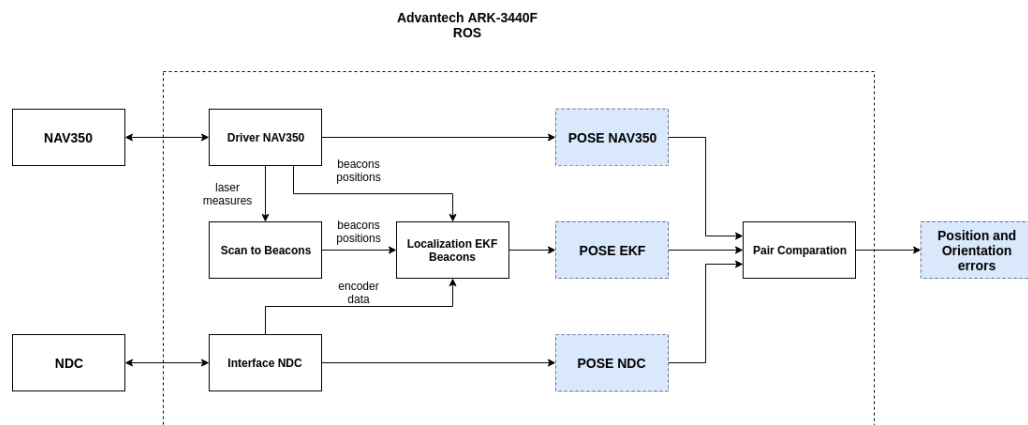


Figura 3.10: Arquitetura do sistema

3.5.2 RVIZ

O RVIZ é uma ferramenta de visualização bastante poderosa disponível no ROS que permite a visualização 3D de dados de sensores e informações de estado daquilo que está a ser executado no ROS. É bastante útil porque possibilita verificar visualmente o estado e o funcionamento do sistema.

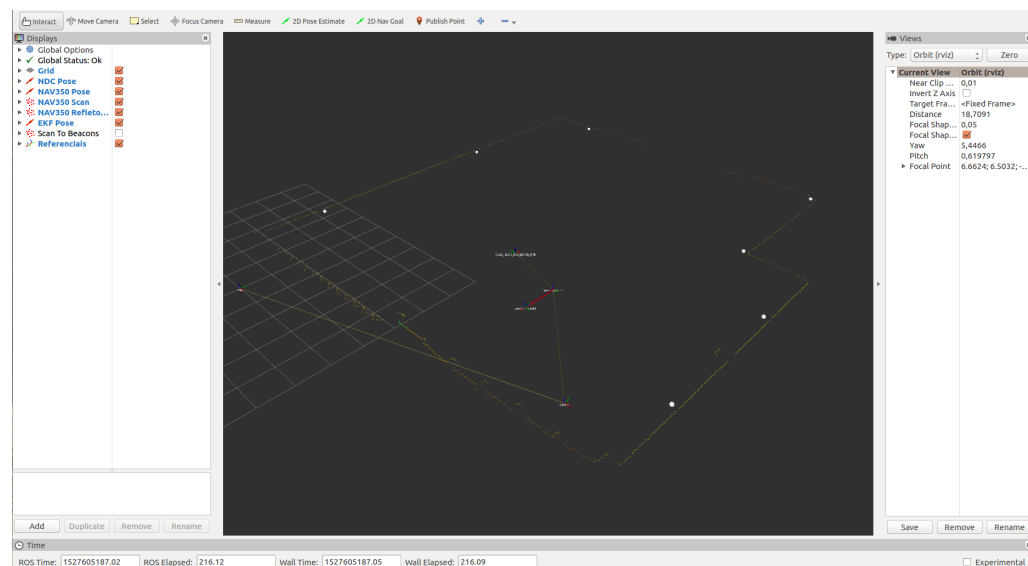


Figura 3.11: RVIZ

Como se pode ver na figura 3.11 é possível subscrever os Nós de interesse e visualizar a sua informação em duas ou três dimensões. Trata-se de uma ferramenta bastante útil para verificar o estado a cada momento do sistema.

3.6 Matlab

O Matlab é uma linguagem de programação de alto nível e ao mesmo tempo um ambiente interativo para computação numérica, visualização e programação. Neste ponto não explicarei o funcionamento do Matlab pois é uma ferramenta largamente utilizada e já bastante conhecida, mas explicarei o porquê de a ter usado nesta tese.

O Matlab permite desenvolver muito facilmente programas para testar aquilo que posteriormente poderá ser implementado no ROS. Mas além disso, as últimas versões do Matlab permitem a conexão ao ROS e com isso é possível validar os desenvolvimentos feitos até então e escritos numa linguagem mais simples. Esta possibilidade é de especial relevância pois o algoritmo de localização global proposto nesta tese foi inicialmente desenvolvido em Matlab e por isso pode ser facilmente testado e alterado sem que tenha de ser constantemente convertido de linguagem Matlab para *C++* ou *Python*. Além disso, com esta nova funcionalidade do Matlab, é com facilidade que se podem analisar dados de Bags de testes que são feitos ao longo da tese. *Bags* são *Logs* de dados que são registados ao longo dos testes.

Capítulo 4

Algoritmo de Localização Global para sistema de localização com refletores indistinguíveis

Neste capítulo será apresentado um algoritmo de determinação da localização de um robô, tendo em consideração o problema de localização global. Este algoritmo é direcionado para robôs que utilizem sensores que medem a distância e ângulo na detecção de refletores artificiais indistinguíveis entre si, instalados no ambiente. Neste caso será exemplificado com um laser rotativo. Assim, será feita uma explicação detalhada do algoritmo proposto no artigo e das funções implementadas para o mesmo. Este algoritmo é baseado no algoritmo de localização global proposto em [\(Ronzoni et al., 2011b\)](#).

4.1 Apresentação do problema

O sistema de localização de um Veículo Autonomamente Guiado é uma das partes fundamentais para o correto funcionamento do mesmo. É a partir desse sistema que o robô determina a sua posição e é capaz de se posicionar corretamente no ambiente onde se encontra. No entanto, podem existir situações em que o robô deixa de saber a sua localização e necessita de despoletar algum mecanismo para voltar a calcular a mesma. Estas situações designam-se habitualmente por "Problema de Localização Global". Entende-se "Localização Global" como a posição do robô nas coordenadas do referencial global. É importante existirem mecanismos que solucionem o problema de localização global, pois aumentam a robustez do sistema de navegação do robô.

4.2 Solução Proposta

a fim de determinar a localização global do robô é apresentado um algoritmo que irá fazer *matching* entre as distâncias dos refletores observados pelo *laser scanner* com as distâncias entre

os refletores no mapa global e calcular a sua posição. Além disso, será também apresentado um algoritmo de filtragem de *outliers*.

O algoritmo que irá ser apresentado é aplicável a robôs móveis que utilizem um sensor que mede as distâncias a balizas (refletores no nosso caso) como tecnologia de localização, assim como refletores artificiais indistinguíveis entre si, instalados no ambiente onde o robô irá operar. Este algoritmo parte do facto de que as distâncias entre refletores no referencial global do mapa são iguais às distâncias entre os refletores observados no referencial do *laser*, a menos do erro das medições do *laser*.

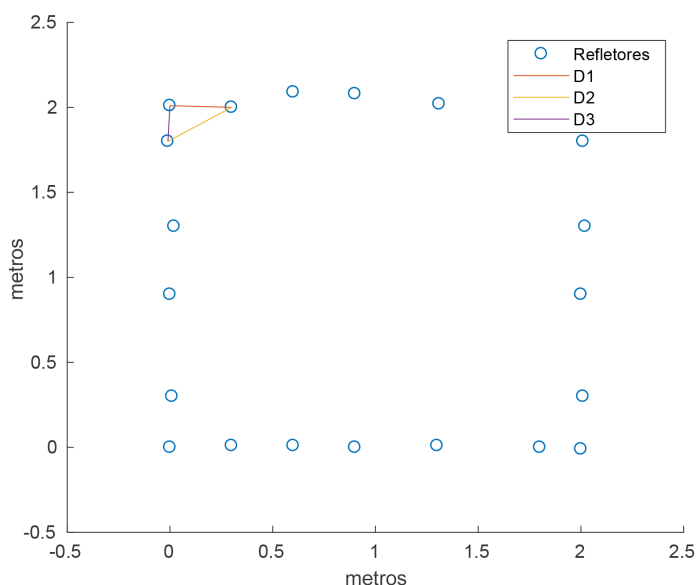


Figura 4.1: Mapa de refletores representado no referencial global com três distâncias entre refletores identificadas, D1, D2 e D3. As distâncias observadas entre refletores é invariante em relação à pose do robô e ao referencial na qual estão a ser observadas.

Na figura 4.1, é possível observar um conjunto de refletores representado no referencial global do mapa, onde o robô se irá movimentar. Para além dos refletores estão representadas três distâncias entre três refletores, são elas D1, D2 e D3. Por sua vez, na figura 4.2, estão representados os refletores detetados pelo laser do robô observados no referencial do mesmo. Para além disso, estão também representadas as distâncias entre os três refletores observados, são elas D4, D5 e D6. Apesar de os refletores estarem a ser observados em referenciais diferentes, as distâncias entre cada um deles é independente do referencial onde estão representados. Ou seja, D1 será igual a D4, D2 será igual a D5 e por fim D3 será igual a D6. Como foi dito anteriormente, este facto foi o mote para o algoritmo apresentado. Além disto, e como foi referido anteriormente, este algoritmo teve por base o algoritmo apresentado em (Ronzoni et al., 2011b).

Tendo em conta isto o objetivo do algoritmo será o de tentar fazer o *matching* entre as distâncias observadas no referencial do *laser* com as distâncias conhecidas *à priori* entre os refletores do

mapa, de forma a identificar quais os refletores que estão a ser observados e com essa informação calcular a posição do robô.

Para além do facto das distâncias entre os refletores serem independentes do referencial onde estão representadas é tido em conta que, de modo a que o algoritmo convirja, terão de ser observados pelo menos três refletores que não estejam equidistantes. Pois esta situação origina três possibilidades de *matching*. No caso de haverem apenas duas observações, levará a que ocorra uma ambiguidade na identificação do refletor correspondente. Para que não haja essa ambiguidade, será necessária a existência da observação de pelo menos mais um refletor e que o conjunto destes três refletores não estejam equidistantes. No entanto, mesmo que estas situações aconteçam e se houverem refletores observados disponíveis, adicionar-se-á mais um refletor observado até que o algoritmo convirja.

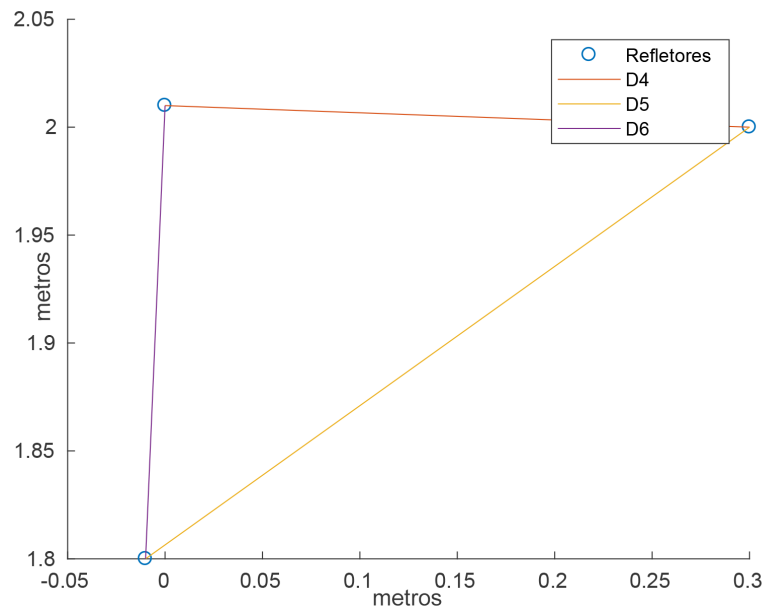


Figura 4.2: Refletores observados pelo robô no referencial do mesmo com três distâncias entre refletores identificadas, D4, D5 e D6. As distâncias observadas entre refletores é invariante em relação à pose do robô e ao referencial na qual estão a ser observadas.

No entanto, para que o algoritmo possa ser aplicado num robô real terá de ser implementado um mecanismo para a filtragem de *outliers*. A todos os possíveis refletores observados pelo *laser* do robô é adicionada a probabilidade de este ser um *outlier*. Após ser encontrada a correspondência entre os pontos observados pelo *laser* do robô e o mapa, irá ser determinada a transformação entre ambos os referenciais. Com essa transformação e com a utilização de mais um refletor observado, que não seja um dos já utilizados para fazer o *matching*, irá ser-lhe aplicada a transformação e verificado se esse ponto corresponde a um refletor no referencial do mapa. Se ao aplicarmos a transformação, o ponto obtido corresponder à posição de um refletor no mapa, a posição do robô foi encontrada. O caso em que esse ponto não corresponde à localização de um refletor no mapa, significa que existem *outliers* nos pontos utilizados para fazer o *matching*. Quando isso acontece,

à probabilidade dos pontos utilizados para fazer *matching* serem um *outliers* será somado um valor constante. A lista destes ponto será ordenada de forma descendente e será feito novamente o processo de *matching*.

A seguir será feita primeiramente a explicação detalhada do algoritmo de *matching* e das funções implementadas no mesmo, ainda sem a parte de filtragem de *outliers* que será explicada mais à frente.

4.3 Algoritmo de *Matching* sem *Outliers*

A seguir é apresentado o algoritmo de *Matching* entre as distâncias calculadas a partir das observações do *laser* do robô e as distâncias entre refletores, calculadas em *offline*, utilizando-se o mapa de refletores.

Algoritmo 1: Algoritmo de *Matching* sem *outliers*

Entrada: Matrizes N , M e $Ndist$

```

1   $Z = \text{maxdist}(M)$ ;
2   $F = f(Z, Ndist, N)$ ;
3  if  $\#F == 1$  then
4  |   return  $F$ ;
5  else
6  |   while  $\#F > 1$  do
7  |   |   if  $\#Z == \#M$  then
8  |   |   |   return  $NULL$ ;
9  |   |   end
10 |   |    $P = \text{select}(M, Z)$ ;
11 |   |    $Z = \text{append}(Z, P)$ ;
12 |   |    $Zdist = \text{dist}(Z)$ ;
13 |   |    $F = \text{match}(Z, Zdist, N, Ndist)$ ;
14 |   |   if  $\#F == 1$  then
15 |   |   |   return  $F$ ;
16 |   |   end
17 |   end
18 end

```

Antes de o algoritmo ser executado é necessário ter disponíveis duas matrizes, são elas a matriz $N \in \{nR \times 2\}$ e $Ndist \in \{nR \times nR\}$ que representam respetivamente as coordenadas x_i e y_i dos refletores representadas no mapa e as distâncias entre todos os refletores. nR representa o número de refletores instalados.

id	0	1	2	...	nR-1
0	$\overline{d00}$	$\overline{d01}$	$\overline{d02}$...	$\overline{d0(nR-1)}$
1	$\overline{d10}$	$\overline{d11}$	$\overline{d12}$...	$\overline{d1(nR-1)}$
2	$\overline{d20}$	$\overline{d21}$	\ddots	...	$\overline{d2(nR-1)}$
...	\vdots	\vdots	\vdots	\ddots	\vdots
nR-1	$\overline{d(nR-1)0}$	$\overline{d(nR-1)1}$	$\overline{d(nR-1)2}$...	$\overline{d(nR-1)(nR-1)}$

Tabela 4.1: Matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.

A matrix N está organizada da seguinte forma:

$$N = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{nR-1} & y_{nR-1} \end{bmatrix} \quad (4.1)$$

Que como foi dito anteriormente, x_i e y_i representam a posição do refletor com o índice i no referencial do mapa. Já a matriz $Ndist$ está organizada da seguinte maneira:

As distâncias representadas na tabela 4.1, como por exemplo $\overline{d02}$, representa a distância entre o refletor com identificador 0 e o refletor com identificador 2, que é calculada a partir da distância euclidiana da seguinte forma:

Para os pontos $P1=(p1x, p1y)$ e $P2=(p2x, p2y)$, a distância euclidiana é dada como:

$$\overline{d12} = \sqrt{(p1x - p2x)^2 + (p1y - p2y)^2} \quad (4.2)$$

A matriz $Ndist$ será uma matriz simétrica, porque $\overline{dij} = \overline{dji}$ e além disso, os elementos da sua diagonal serão iguais a 0, porque representam a distância entre o mesmo ponto. Uma vez que a localização dos refletores no referencial do mapa é constante, a menos que haja uma alteração do *layout* da fábrica, a matriz $Ndist$ pode ser calculada em *offline* e com isso poupar recursos de processamento durante a fase *online*. Da mesma forma, a matriz N com as posições dos refletores no referencial do mapa, é previamente preenchida.

Como entrada do algoritmo há também a matriz M que corresponde às posições (x,y) no referencial do *laser* do robô dos possíveis refletores detetados. Neste momento o número mínimo de pontos na matriz M para que não existam ambiguidades ainda não é tido em conta, mas mais à frente irá ser introduzido essa característica.

Na linha 1 do algoritmo 1, este começa por seleccionar os dois pontos que se encontram à maior distância, dentro do conjunto de pontos M , com a função $Z=maxdist(M)$. A matriz Z é organizada da seguinte forma:

$$Z = \begin{bmatrix} p1x & p1y \\ p2x & p2y \end{bmatrix} \quad (4.3)$$

Em que, $P1=(p1x, p1y)$ e $P2=(p2x, p2y)$ são os dois pontos pertencentes a M que estão à maior distância um do outro. De seguida o algoritmo irá usar a distância entre esses dois pontos para tentar fazer *matching* com todas as distâncias entre cada um dos refletores do mapa. Ou seja, irá ser feita uma busca na matriz $Ndist$ por distâncias idênticas à distância entre $P1$ e $P2$. Para isso, na linha 2 do algoritmo 1 é chamada a função $F = f(Z, Ndist, N)$. Esta função irá fazer uma busca em toda a matriz $Ndist$ por distâncias idênticas à distância entre $P1$ e $P2$. Como as medidas do sensor têm um erro associado, é feita uma comparação entre a distância entre $P1$ e $P2$ com todas as distâncias da matriz $Ndist$ com uma dada tolerância associada. O retorno desta função F serão as coordenadas dos pares de pontos de refletores, no referencial do mapa, que se encontram a uma distância idêntica à distância entre $P1$ e $P2$. A matriz $F \in (nF \times 4)$ está organizada da seguinte forma:

$$F = \begin{bmatrix} x1 & y1 & x1' & y1' \\ x2 & y2 & x2' & y2' \\ \vdots & \vdots & \vdots & \vdots \\ xnF & ynF & xnF' & ynF' \end{bmatrix} \quad (4.4)$$

Em que nF corresponde ao número de correspondências encontradas em $Ndist$. Cada linha da matriz F tem as coordenadas (xi, yi) e (xi', yi') dos dois pontos cuja distância é idêntica à procurada. Se nF for igual a um, significa que apenas uma correspondência foi encontrada e o algoritmo deverá parar aqui. Usar apenas dois pontos não levará a uma solução, pois, ao usarmos apenas uma distância entre esses dois pontos, haverá sempre a ambiguidade em sabermos qual dos pontos corresponde a qual. Tal situação pode ser visualizado na figura 4.3. Apesar da distância entre os dois refletores ter apenas uma correspondência, não se sabe se essa distância foi observada na posição 1 ou na posição 2.

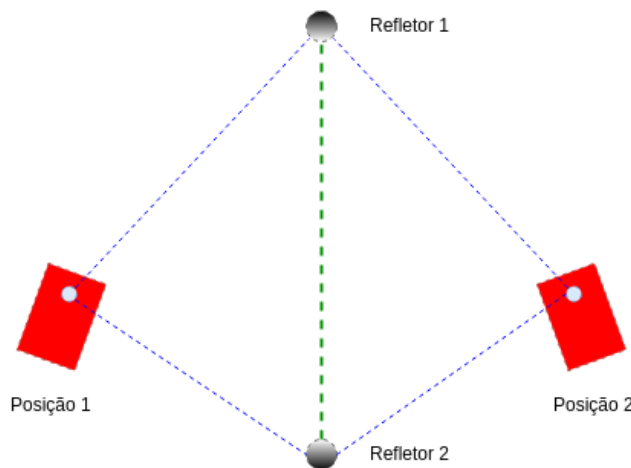


Figura 4.3: Exemplo ilustrativo da ambiguidade que acontece quando se usa apenas uma distância para fazer *matching*

id	0	1	...	nZ-1
0	$\overline{d00}$	$\overline{d01}$...	$\overline{d0(nZ-1)}$
1	$\overline{d10}$	$\overline{d11}$...	$\overline{d1(nZ-1)}$
...	\vdots	\vdots	\ddots	\vdots
nZ-1	$\overline{d(nZ-1)0}$	$\overline{d(nZ-1)1}$...	$\overline{d(nZ-1)(nZ-1)}$

Tabela 4.2: Matriz $Zdist$ com as distâncias entre cada um dos refletores do mapa.

No entanto, mais à frente irá ser apresentado um mecanismo para resolver esta ambiguidade, com a utilização de um terceiro ponto do conjunto M . Caso existam várias correspondências, ou seja nF seja maior que um, o algoritmo irá entrar num ciclo, linha 6. Neste ciclo é feita primeiramente a verificação se se estão a utilizar todos os pontos da matriz M , linha 7. Se isso acontecer e se nF for maior que um, significa que o algoritmo não irá convergir e serão necessárias novas medidas. Se ainda não se estão a utilizar todos os pontos de M é selecionado um ponto extra de M que não sejam os pontos já presentes em Z , com a função $P=select(M,Z)$, na linha 10. Esse ponto, P , à matriz Z , na linha 11, com a função $Z=append(Z,P)$.

Na linha 12 é calculada a matriz $Zdist$ de distâncias entre cada um dos pontos do conjunto Z . A matriz tem a organização apresentada em 4.2. Tal como a matriz $Ndist$, esta matriz também é simétrica e os valores da diagonal são igual a zero.

Na linha 13 do algoritmo 1, irá ser feito o *matching* com a nova lista de pontos Z . Se for encontrada apenas uma correspondência entre as distâncias entre os pontos de Z e as distâncias entre cada refletor do mapa em $Ndist$, o algoritmo para e retorna as posições dos refletores correspondentes, no referencial do mapa. Caso contrário, irá ser feita mais um iteração do ciclo, onde será feita a verificação se já se estão a utilizar todos os pontos da observação e adicionado mais um ponto ao conjunto Z para tentar fazer *matching*.

4.3.1 Função match

Esta função irá procurar as distâncias de $Zdist$ na matriz $Ndist$ e retornar os pontos se houverem correspondências. Esta procura irá ter em conta as conexões entre os pontos utilizados. Por exemplo, se o meu conjunto Z tiver três pontos, $P1$, $P2$ e $P3$ e consequentemente três distâncias entre eles, $\overline{d12}$, $\overline{d23}$ e $\overline{d31}$, que conectam os três pontos. Ao encontrar uma correspondência da distância $\overline{d12}$ em $Ndist$, entre um ponto $P1'$ e $P2'$, a distância $\overline{d23}$ terá de corresponder a uma distância entre $P2'$ e outro ponto qualquer. Isto, permite que, após procurar a primeira distância em toda a matriz $Ndist$ e encontrar as suas correspondências, a segunda distância já não será procurada em toda a matriz $Ndist$ mas em apenas algumas linhas de $Ndist$, correspondentes às distâncias entre $P2'$ a outro ponto, neste caso poderia ser a distância entre $P2'$ e $P3'$. Por fim, a ultima distância, $\overline{d31}$ será procurada, já não em linhas mas em pontos específicos da matriz $Ndist$. Neste exemplo, a terceira distância, $\overline{d31}$ terá de ser igual à distância entre $P3'$ e $P1'$ e portanto estará na posição $(3,1)$ da matriz $Ndist$.

Existirão três etapas de procura, uma que irá percorrer toda a matriz uma única vez, uma segunda que irá percorrer apenas determinadas linhas e uma terceira que irá verificar apenas pontos específicos da matriz. A seguir irá ser apresentado um exemplo para uma melhor compreensão do processo de procura.

id	0	1	2
0	0.0	9.34	11.28
1	9.34	0.00	5.01
2	11.28	5.01	0.00

Tabela 4.3: Exemplo de matriz $Zdist$ com as distâncias entre cada um dos refletores observado.

id	0	1	2	3	4	5	6	7
0	0.00	1.14	5.73	0.35	9.34	11.35	9.34	11.28
1	1.14	0.00	9.03	8.63	6.69	8.26	1.12	7.52
2	5.73	9.03	0.00	5.80	2.12	5.80	9.49	9.28
3	0.35	8.63	5.80	0.00	2.78	5.46	9.48	5.26
4	9.34	6.69	2.12	2.78	0.00	0.69	8.38	5.01
5	11.35	8.26	5.80	5.46	0.69	0.00	1.73	9.57
6	9.34	1.12	9.49	9.48	8.38	1.73	0.00	6.01
7	11.28	7.52	9.28	5.26	5.01	9.57	6.01	0.00

Tabela 4.4: Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.

Se o conjunto Z for composto por três pontos $P0$, $P1$ e $P2$ em que as distâncias entre esses pontos sejam, $\overline{d01} = 9.34$, $\overline{d12} = 5.01$ e $\overline{d21} = 11.28$, representadas na matriz 4.3, a função irá começar por correr uma vez toda a matriz $Ndist$, representada na tabela 4.4 à procura de um valor igual a 9.34. Neste caso exemplificativo, trata-se de uma situação ideal, sem erros. No entanto, na prática existirá uma dada tolerância para se fazerem estas correspondências, pois as medidas vindas do laser têm um erro associado. Ao percorrer a matriz irão ser encontrados quatro valores iguais a 9.34 correspondentes às distâncias $\overline{d04}$, $\overline{d06}$, $\overline{d40}$ e $\overline{d60}$, como podemos verificar na tabela 4.5.

id	0	1	2	3	4	5	6	7
0	0.00	1.14	5.73	0.35	9.34	11.35	9.34	11.28
1	1.14	0.00	9.03	8.63	6.69	8.26	1.12	7.52
2	5.73	9.03	0.00	5.80	2.12	5.80	9.49	9.28
3	0.35	8.63	5.80	0.00	2.78	5.46	9.8	5.26
4	9.34	6.69	2.12	2.78	0.00	0.69	8.38	5.01
5	11.35	8.26	5.80	5.46	0.69	0.00	1.73	9.57
6	9.34	1.12	9.49	9.48	8.38	1.73	0.00	6.01
7	11.28	7.52	9.28	5.26	5.01	9.57	6.01	0.00

Tabela 4.5: Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.

O próximo passo será pesquisar nas linhas 4, 6 e 0 a distância $\overline{d12} = 5.01$. Ao fazer a busca nas linhas 4, 6 e 0, encontramos a distância 5.01 que neste caso irá corresponder à distância $\overline{d47}$, como pode ser verificado na tabela 4.6

id	0	1	2	3	4	5	6	7
0	0.00	1.14	5.73	0.35	9.34	11.35	9.34	11.28
1	1.14	0.00	9.03	8.63	6.69	8.26	1.12	7.52
2	5.73	9.03	0.00	5.80	2.12	5.80	9.49	9.28
3	0.35	8.63	5.80	0.00	2.78	5.46	9.48	5.26
4	9.34	6.69	2.12	2.78	0.00	0.69	8.38	5.01
5	11.35	8.26	5.80	5.46	0.69	0.00	1.73	9.57
6	9.34	1.12	9.49	9.48	8.38	1.73	0.00	6.01
7	11.28	7.52	9.28	5.26	5.01	9.57	6.01	0.00

Tabela 4.6: Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.

Por fim, e como foi dito anteriormente, a última etapa da busca será em pontos específicos. Ora, como $\overline{d01} = 9.34$ (matriz $Zdist$), corresponde à distância $\overline{d04}$ (matriz $Ndist$), $\overline{d12} = 5.01$ (matriz $Zdist$) corresponde à distância $\overline{d47}$ (matriz $Ndist$), obrigatoriamente $\overline{d21} = 11.28$ (matriz $Zdist$) irá ter de corresponder à distância $\overline{d07}$ (matriz $Ndist$), como podemos ver na tabela 4.7 .

id	0	1	2	3	4	5	6	7
0	0.00	1.14	5.73	0.35	9.34	11.35	9.34	11.28
1	1.14	0.00	9.03	8.63	6.69	8.26	1.12	7.52
2	5.73	9.03	0.00	5.80	2.12	5.80	9.49	9.28
3	0.35	8.63	5.80	0.00	2.78	5.46	9.48	5.26
4	9.34	6.69	2.12	2.78	0.00	0.69	8.38	5.01
5	11.35	8.26	5.80	5.46	0.69	0.00	1.73	9.57
6	9.34	1.12	9.49	9.48	8.38	1.73	0.00	6.01
7	11.28	7.52	9.28	5.26	5.01	9.57	6.01	0.00

Tabela 4.7: Exemplo de matriz $Ndist$ com as distâncias entre cada um dos refletores do mapa.

Assim, podemos dizer que os refletores nos pontos $P0$, $P1$ e $P2$, no referencial do *laser* do robô, correspondem aos pontos $P0$, $P4$ e $P7$ no referencial global. Tendo os índices dos pontos, podemos ir buscar diretamente as coordenadas desses pontos à matriz N .

4.4 Algoritmo de *Matching* com *Outliers*

Sem um mecanismos de filtragem de *outliers* o algoritmo não terá robustez e não irá convergir corretamente em muitos casos. Por isso, irá ser apresentado um mecanismo de filtragem de *outliers* baseado no artigo (Ronzoni et al., 2011b) mas com algumas alterações em relação a esse.

O algoritmo de *Matching* com *outliers* irá englobar o algoritmo apresentado anteriormente sem *outliers*. A parte apresentada anteriormente sem *outliers* está representada aqui pela função $[F, Z] = matching(N, K, Ndist)$, na linha 5.

Para além das entradas N (posições dos refletores), M (observações no referencial do robô) e $Ndist$ (matriz de distâncias entre cada refletor do mapa), existe também um fator k . Este fator corresponde ao número mínimo de posições de supostos refletores observados, que serão utilizados para fazer *matching*.

Na linha 1 do algoritmo é atribuída um "peso" a cada observação de um possível refletor. Esse valor é inicialmente colocado igual a zero. Daqui resulta a matriz W , representada na tabela 4.9, que é igual à matriz de observações M , representada na tabela 4.8, mas com uma coluna extra onde estão guardados os "pesos" de cada observação.

Algoritmo 2: Algoritmo de *Matching* com outliers

Entrada: Matrices N, M, Ndist e k

```

1 W=setweight(M,0);
2 while #RESULT == 0 do
3   S = sortrows(W);
4   K = choose(S, k);
5   [F, Z] = matching(N,K,Ndist);
6   if #F == 1 then
7     [ RESULT, R, t ] = associate( Z, N, F );
8     if (#k < #nS) and RESULT then
9       P=select(S,K);
10      CONFIRM = confirmation(N,P,R,t);
11      if CONFIRM then
12        return R, t
13      else
14        W = incweight( W, K,  $\gamma$  );
15      end
16    end
17  else
18    W = incweight( W, K,  $\gamma$  );
19  end
20 end

```

id	x	y
0	x0	y0
1	x1	y1
\vdots	\vdots	\vdots
nM-1	xnM-1	ynM-1

Tabela 4.8: Exemplo de matriz M com as posições observados de cada possível refletor no referencial do *laser*.

id	x	y	weight
0	x0	y0	0
1	x1	y1	0
\vdots	\vdots	\vdots	0
nM-1	xnM-1	ynM-1	0

Tabela 4.9: Exemplo de matriz W com as posições observados de cada possível refletor no referencial do *laser* e os pesos associados a cada uma delas.

A seguir, o algoritmo entra num ciclo enquanto não convergir para uma solução, linha 2. Dentro desse ciclo começa-se por ordenar as observações, de forma descendente, de acordo com os pesos atribuídos, linha 3. Daqui resulta a matriz S , que tem uma estrutura igual à W , com a única diferença de estar ordenada de forma descendente. Depois de as observações estarem ordenadas de forma descendente, são escolhidas as k observações com menor peso e guardadas em K , que à semelhança com a matriz W e S , tem a mesma estrutura delas.

O passo seguinte, na linha 4, será o de chamar a função *matching* que corresponde ao algoritmo 2. Aqui, como já foi explicado antes, irá tentar ser encontrada uma correspondência única entre as distâncias dos supostos refletores vistos no referencial do *laser* com as distâncias entre todos os refletores, no referencial do mapa.

Se for encontrada uma única correspondência, linha 6, o algoritmo irá começar por calcular a transformação entre os referenciais das observações, ou seja, no referencial do *laser* e o referencial do mapa, linha 7 e verificar se as observações, após lhe ser aplicada essa transformação, correspondem a pontos do mapa. Para isso é executada a função *associate*, que tem como parâmetros de entrada os pontos Z , utilizados para fazer o *matching*, os índices dos pontos correspondentes no referencial do mapa F e o mapa N . Como saídas da função teremos um booleano *RESULT*, que se tiver o valor 1, significa que os pontos usados para encontrar a transformação correspondem a pontos de refletores do mapa, a matriz R de rotação, e o vetor t de translação.

Após calculadas a matriz de rotação R e o vetor de translação t , se houverem pontos das observações disponíveis, linha 8, é selecionado um ponto das observações, linha 9, que não os utilizados para fazer *matching* ou achar a transformação entre referenciais para fazer a confirmação tanto da transformação como consequentemente do *matching*. É possível que um dos pontos usados para fazer *matching* e posteriormente para achar a transformação entre referenciais seja um *outlier*, por

isso, na linha 10, na função *confirmation* é aplicada a transformação a esse ponto e verificado se corresponde à posição de um refletor no mapa. Se isso acontecer a variável CONFIRMATION tomará o valor 1, caso contrário tomará o valor 0. Ora, se CONFIRMATION for igual a 1 pode retornar-se a transformação, se CONFIRMATION tiver o valor 0, significa que um dos pontos utilizados será um *outlier* e por isso é lhe somada um valor constante γ , que tem como significado o peso de poder ser um *outlier*. Esse incremento também acontece quando não é encontrada uma correspondência única ao chamarmos a função de *matching*.

4.4.0.1 Função *associate*

Esta função está dividida em três etapas. A primeira é bastante simples e consiste em obter as coordenadas dos pontos no referencial do mapa através dos seus identificadores. A segunda consiste em obter a matriz de rotação e de translação, da transformação entre o referencial do *laser* e o referencial do mapa e por fim, a terceira etapa consiste em verifica de, ao aplicarmos a transformação às observações, estas coincidem com pontos de refletores do mapa.

O grande desafio nesta função é a segunda etapa, de descobrir a transformação entre referenciais. Tendo em conta o artigo (Sorkine, 2009), podemos encarar este problema como a procura de uma transformação constituída por uma matriz de rotação R e um vetor de translação t que minimize o erro quadrático médio entre dois conjuntos de pontos em R^d $P=(p1, p2, ..., pn)$ e $Q=(q1, q2, ..., qn)$. Ou seja:

$$(R, t) = \underset{R \in SO(d), t \in R^d}{\operatorname{argmin}} \sum_{i=1}^n w_i \|(R * p_i + t) - q_i\|^2 \quad (4.5)$$

O primeiro passo será o cálculo das médias ponderadas dos dois conjuntos de pontos, através de:

$$\bar{p} = \frac{\sum_{i=1}^n w_i * p_i}{\sum_{i=1}^n w_i} \quad (4.6)$$

e

$$\bar{q} = \frac{\sum_{i=1}^n w_i * q_i}{\sum_{i=1}^n w_i} \quad (4.7)$$

Após serem calculados os "centróides" dos dois conjuntos de pontos, estes serão subtraídos a cada ponto dos respetivos conjuntos. Ouse seja:

$$x_i := p_i - \bar{p}, \quad y_i := q_i - \bar{q}, \quad i = 1, 2, \dots, n \quad (4.8)$$

O passo seguinte é o calculo da matriz de covariancias de dimensão $d \times d$:

$$S = XWY^T \quad (4.9)$$

onde Z e Y são as matrizes de dimensão $d \times n$ que contêm x_i e y_i respetivamente e $W = \text{diag}(w_1, w_2, \dots, w_n)$. Após isso é calculada a decomposição em valores singulares de S , em que se obtém $S = USV^T$. A partir daqui pode calcular-se a matriz de rotação através de:

$$R = V * \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \det(VU^T) \end{pmatrix} * U^T \quad (4.10)$$

e calcular a translação através de:

$$t = \bar{q} - R\bar{p} \quad (4.11)$$

Uma vez definidas a matriz de rotação e de translação, estas são aplicadas às observações e é verificado se correspondem a um ponto/refletor do mapa. A posição do robô no referencial global corresponderá ao ponto obtido após aplicar a matriz de rotação e translação ao ponto $(0,0)$ do referencial do robô.

Capítulo 5

Testes e resultados

Neste capítulo será inicialmente descrito o ambiente onde tiveram lugar os testes com o AGV, de seguida irá ser explicado cada um dos testes dos vários sistemas de localização, a comparação entre o sistema de deteção de refletores por parte do NAV350 e dos sistemas desenvolvidos pelo INESC-TEC (*Extended Kalman Filter* e *Perfect Match*), será também analisado o algoritmo de localização com base em contornos *Perfect Match*, o teste com o algoritmo de localização global implementado neste trabalho e por fim o sistema de *Ground Truth*.

Neste trabalho foram usados dois sistemas a funcionarem em simultâneo, o sistema da NDC, que usa o laser de navegação KOLLMORGEN LS5, já instalado originalmente no AGV, e o sistema do INESC-TEC, que usa o laser de navegação SICK NAV350, que foi posteriormente instalado neste AGV, na qual serão testados vários algoritmos de localização para comparação com a localização dada pelo sistema da NDC e também para comparações entre si.

Como foi dito anteriormente, o sistema da NDC funciona como uma caixa negra que irá fornecer uma pose para ser comparada com a pose dada pelo laser SICK NAV350, a pose dada pelo *Extended Kalman Filter* e com a pose dada pelo *Perfect Match*.

Tendo isto serão feitas as seguintes comparações:

- NDC - *Extended Kalman Filter*;
- NDC - Localização SICK NAV350;
- Localização SICK NAV350 - *Extended Kalman Filter*;

Uma vez que o laser de navegação SICK NAV350 também possui a funcionalidade de localização e identificação de refletores é de interesse fazer a comparação entre essa funcionalidade e o sistema anteriormente proposto em (Sobreira, 2017), para identificação, localização e filtragem de *outliers*, usando as medições do laser de navegação. Ou seja:

- Deteção de refletores SICK NAV350 - *Scan to Beacons*;

Por conseguinte e uma vez que o *Extended Kalman Filter* utiliza como entrada os dados com a informação da localização dos refletores é de especial interesse fazer as comparações deste método

com os restantes usando a detecção de refletores do SICK NAV350 e do *Scan to Beacons*. A primeira lista de comparações passará a ser:

- NDC - *Extended Kalman Filter*, com detecção de refletores do SICK NAV350;
- NDC - *Extended Kalman Filter*, com detecção de refletores *Scan to Beacons*;
- NDC - Localização SICK NAV350;
- Localização SICK NAV350 - *Extended Kalman Filter*, com detecção de refletores do SICK NAV350;
- Localização SICK NAV350 - *Extended Kalman Filter*, com detecção de refletores *Scan to Beacons*;

Todos os algoritmos comparados anteriormente pertencem à mesma família, ou seja, necessitam da detecção de marcos artificiais colocados no ambiente (refletores cilíndricos no nosso caso) para que possam calcular uma dada pose. No entanto, uma vez que têm sido desenvolvidos e melhorados algoritmos de localização baseado em contornos, que não necessitam da instalação de hardware externo e se baseiam nas características naturais do meio para calcularem uma pose, será feita a comparação de um desses algoritmos, o *Perfect Match* com os algoritmos anteriormente comparados. Serão feitas as seguintes comparações:

- *Perfect Match* - NDC;
- *Perfect Match* - Localização SICK NAV350;
- *Perfect Match* - *Extended Kalman Filter*.

Além disso será testado o algoritmo de localização global implementado nesta para estimação da posição inicial do AGV.

5.1 Ambiente de teste

Os testes seguintes tiveram lugar num armazém com a topologia representada na na figura 5.1. A cinza estão representados os locais deste espaço que se encontram ocupados. Apesar de ser um espaço amplo, a dimensão do AGV e a quantidade de espaço ocupado limitaram a possibilidade de trajetórias que este pode executar.

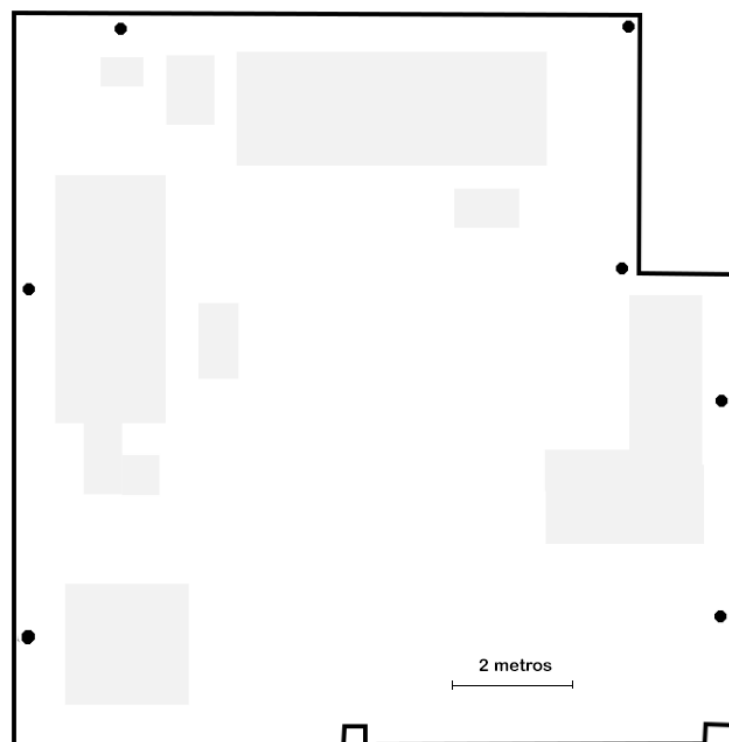


Figura 5.1: Mapa de ocupação do local de testes

5.1.1 Trajetória de testes

Como foi dito anteriormente, a trajetória efetuada pelo AGV foi pré definida pelos técnicos da CONSOVEYO e apenas pode ser alterada por integradores da NDC. A trajetória efetuada pelo robô é a representada na figura 5.2.

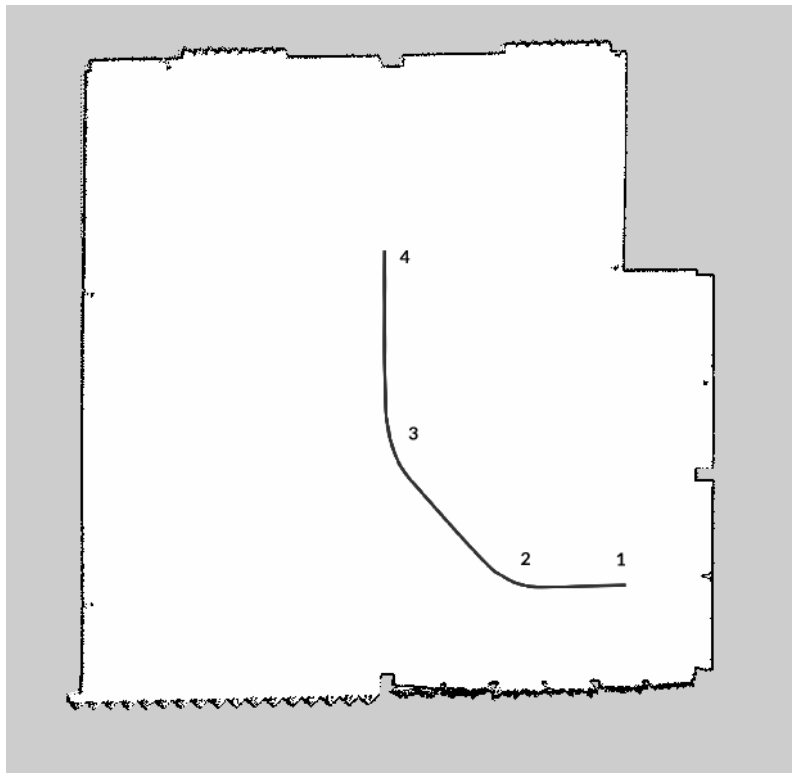


Figura 5.2: Mapa do espaço feito a partir do *package* ROS Hector Mapping com a trajetória de testes que o AGV efetuou

O robô parte do repouso do ponto 1 e segue em linha reta até ao ponto 2, onde executa uma pequena curva à direita, de seguida anda em linha reta até ao ponto 3 onde volta a fazer uma ligeira curva à direita e de seguida vai em linha reta até ao ponto 4, onde para, de seguida executa a trajetória inversa até ao ponto 1.

Os sistemas foram comparados aos pares e a correrem em simultâneo, enquanto o AGV executava a trajetória pré-definida e ao longo desse percurso foram guardados os seguintes dados, de cada um dos sistemas, para comparação:

- Posição do Sistema 1;
- Posição do Sistema 2;
- Orientação do Sistema 1;
- Orientação do Sistema 2;
- Diferença de Posição;
- Diferença de Orientação.

Além disso também foram calculados os seguintes valores, de forma a quantificar as diferenças entre os sistemas.

- Diferença média absoluta de posição;
- Desvio padrão da diferença de posição;
- Diferença máxima de posição;
- Diferença média de orientação;
- Desvio padrão da diferença de orientação;
- Diferença máxima de orientação.

5.2 Comparação da pose calculada pelo NDC com a do NAV350

Como se verifica na figura 5.3, onde se compara a posição calculada pelo sistema da NDC e pelo laser *SICK NAV350*, as posições dadas pelo sistema da NDC descrevem uma trajetória mais suave, comparativamente às posições dadas pelo laser *SICK NAV350*. É possível verificar algumas oscilações na posição dada pelo *SICK NAV350* quando passa pela curva 2 da figura 5.2. Tal está evidenciado na figura 5.4. No entanto a diferença entre a posição dada por um sistema e pelo outro são bastante idênticas.

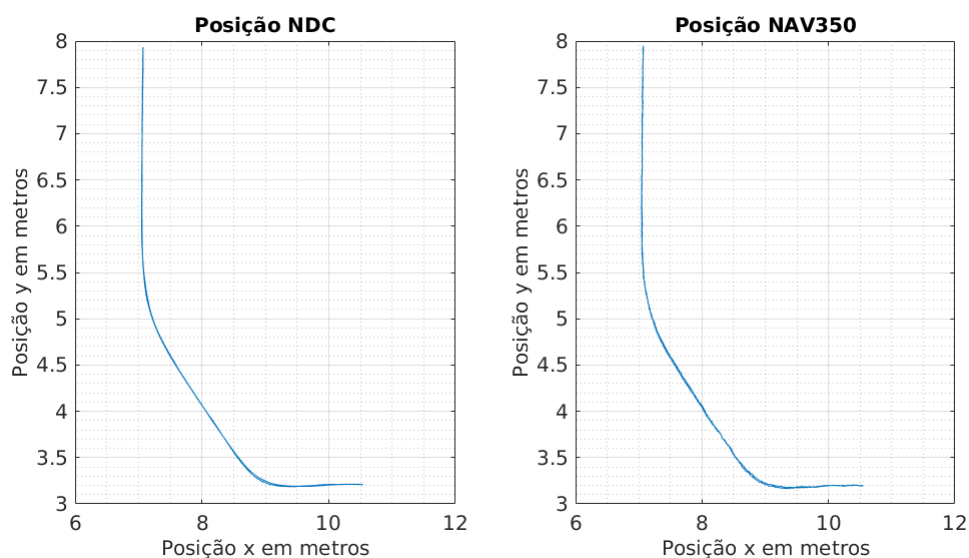


Figura 5.3: Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo NAV350 ao longo da trajetória

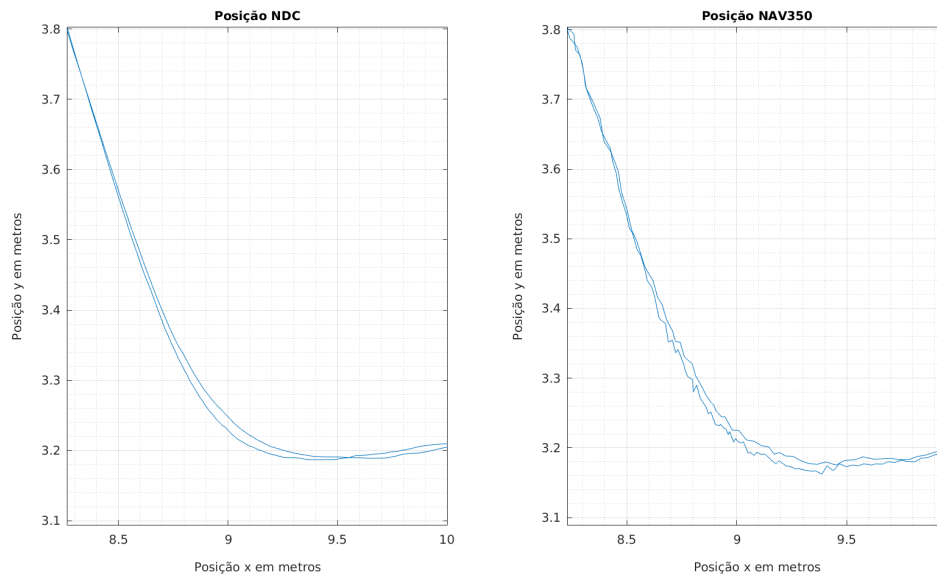


Figura 5.4: Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo NAV350 ao longo da trajetória, onde é evidenciada a curva 2

Na figura 5.5 onde é analisada a diferença de posição entre um sistema e outro pode notar-se que os valores de $t = 20$ s, $t = 30$ s e $t = 70$ s, $t = 80$ s onde a diferença é maior coincidem com os momentos em que o AGV executa as curvas 2 e 3 da figura 5.2 pela primeira e segunda vez, respetivamente.

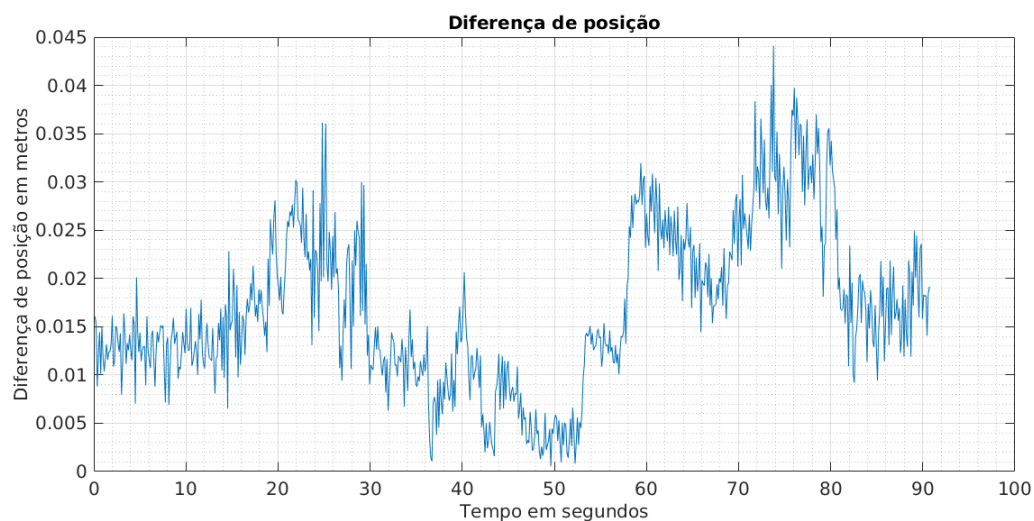


Figura 5.5: Diferença de posição em metros entre posição dada pelo NDC e NAV350 ao longo do teste

Na tabela 5.1 estão representados os valores da diferença média absoluta, máxima e do desvio

padrão calculados a partir dos valores da figura 5.5, onde verificamos que a diferença máxima de posição ocorre aquando da segunda passagem pelo ponto 2 da figura 5.2 e é de cerca de 0.0441 m. No entanto durante as partes do trajeto em linha reta a diferença baixa para cerca de 0.0150 m, sendo que o valor médio da diferença absoluta é de 0.0169 m com um desvio padrão de 0.0083 m.

No que diz respeito à orientação, as diferenças entre um sistema e outro, são difíceis de notar ao analisar apenas a figura 5.6.

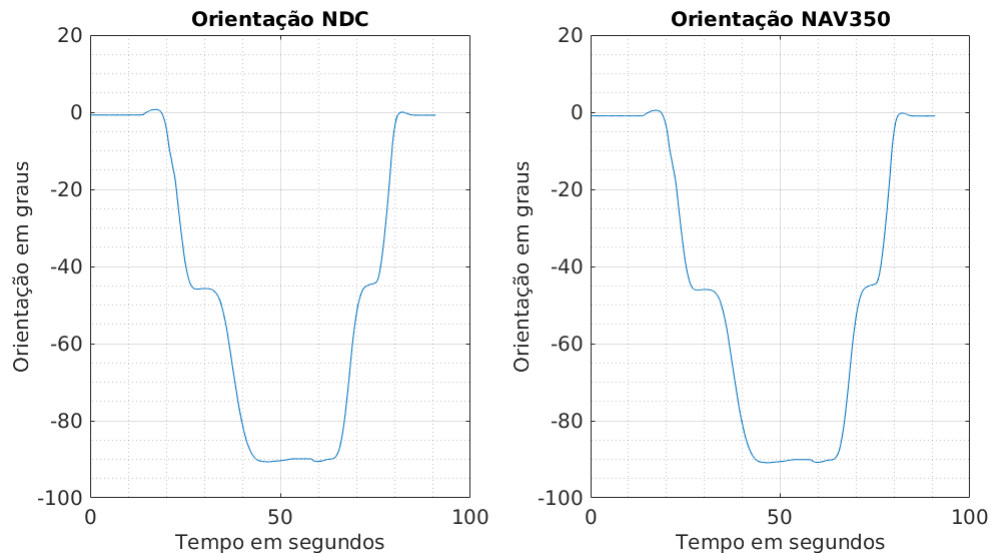


Figura 5.6: Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo NAV350 ao longo da trajetória

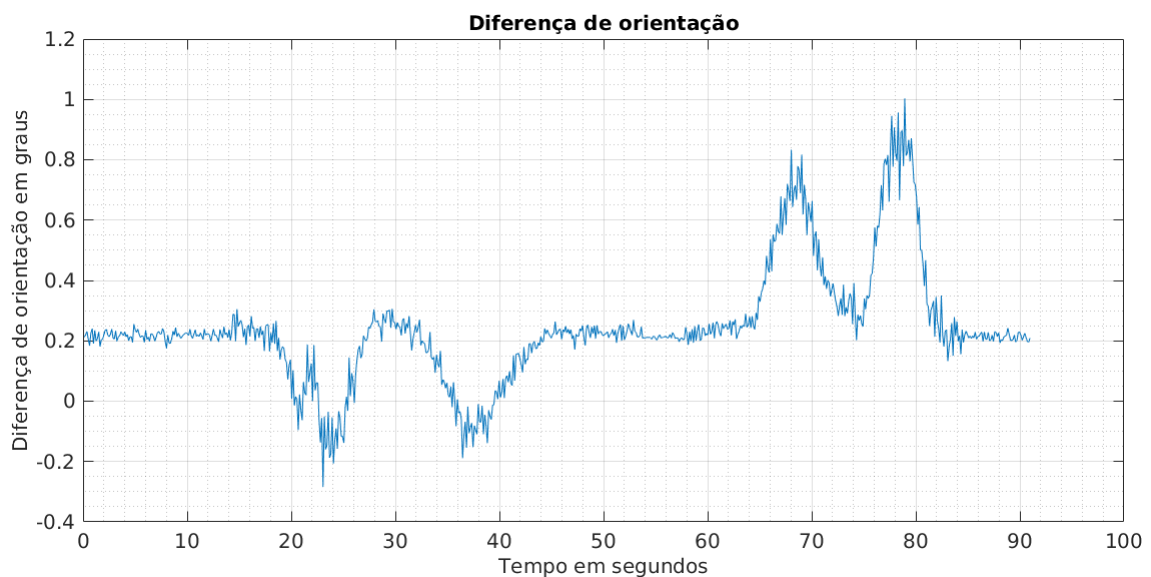


Figura 5.7: Diferença de orientação em graus entre orientação dada pelo NDC e NAV350 ao longo do teste

É possível verificar as diferenças de orientação entre os dois sistemas observando a figura 5.7 e a tabela 5.1. Tal como a posição, os valores onde o erro de orientação é maior no momento onde o AGV executa a curva 2 em $t=$ e $t=$, e a curva 3 em $t=$ e $t=$. Como se verifica na tabela 5.1 o erro máximo de orientação é de 1.0032° . No entanto o erro médio de orientação é mais baixo, 0.2432° com um desvio padrão de 0.1889. É de notar na figura 5.1, que quando o robô executa um movimento retilíneo o erro situa-se à volta de 0.2° . Este valor pode dever-se ao facto de os dois laser de navegação não estarem perfeitamente alinhados e existir um ligeiro *offset* entre ambos. Esta diferença entre a posição dos lasers pode ser futuramente compensada e calibrada.

Diferença absoluta média de posição	0.0169 m
Desvio padrão da diferença de posição	0.0083 m
Diferença máxima de posição	0.0441 m
Diferença média de orientação	0.2432°
Desvio padrão da diferença de orientação	0.1889°
Diferença máxima de orientação	1.0032°

Tabela 5.1: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e NAV350

5.3 NDC vs EKF com deteção de refletores NAV350

Os resultados obtidos nesta comparação foram bastante semelhantes aos obtidos na comparação anterior entre o sistema da NDC e o laser *SICK NAV350*. No entanto e como se pode verificar na figura 5.8, as posições dadas pelo *Extended Kalman Filter* descrevem uma trajetória mais suave e idêntica à dada pelo sistema da NDC. Tal pode ser visto com mais detalhe na figura 5.9.

Mais uma vez, as maiores diferenças de posição ocorrem aquando da passagem pelas curvas assinaladas com os números 2 e 3 na figura 5.2, onde toma o valor máximo de 0.0385 em $t = 86$ s. Tal como acontecia na comparação anterior a diferença tende a baixar durante as trajetórias retilíneas. Apesar da trajetória calculada pelo *Extended Kalman Filter* parecer mais suave que a calculada pelo laser *SICK NAV350*, a diferença média do *Extended Kalman Filter* é ligeiramente superior, ou seja, 0.0172 m, com um desvio padrão de 0.0068, comparativamente com os 0.0169 m obtidos anteriormente.

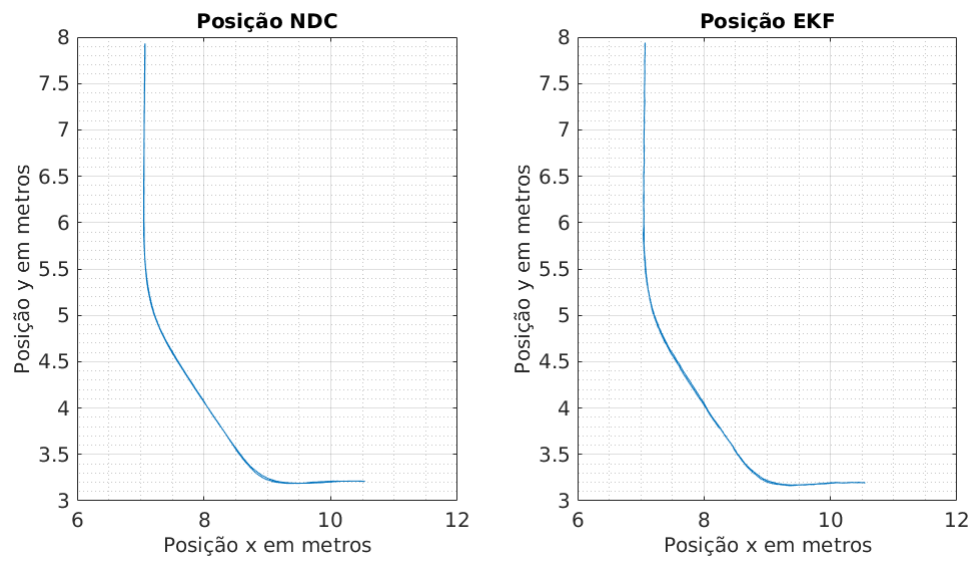


Figura 5.8: Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória

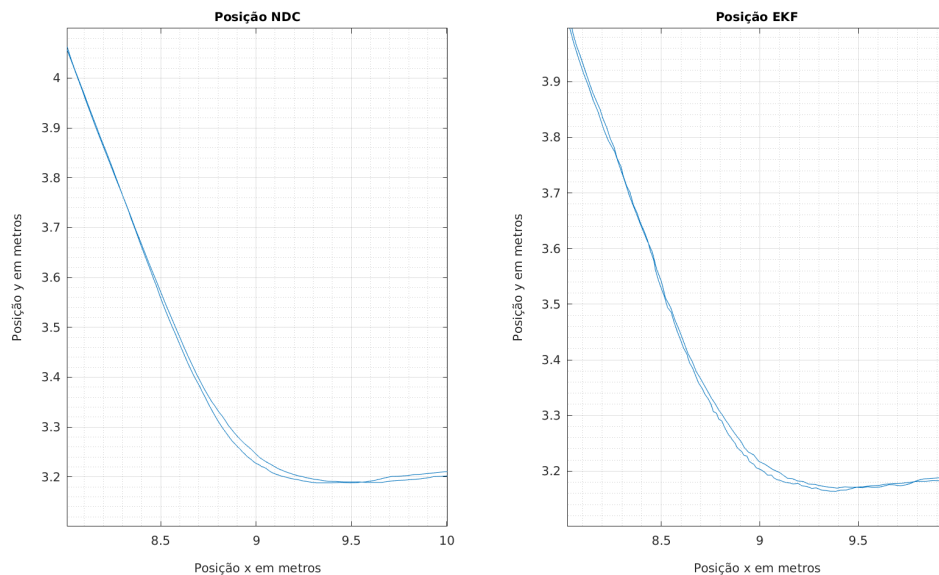


Figura 5.9: Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, onde é evidenciada a curva 2

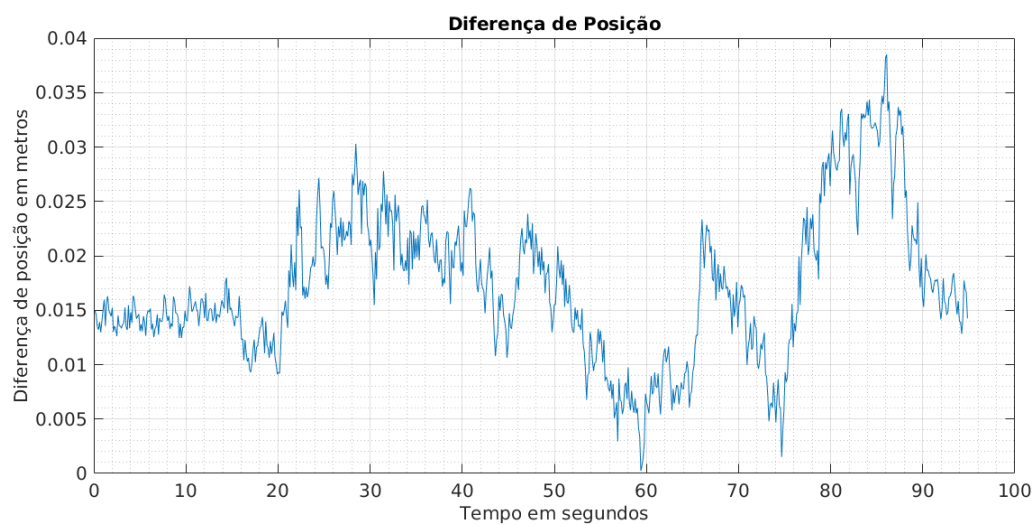


Figura 5.10: Diferença absoluta de posição em metros entre posição dada pelo NDC e EKF ao longo do teste

No que diz respeito à orientação é necessário analisar a figura 5.12 e a tabela 5.3 porque as diferenças de orientação são imperceptíveis ao analisar apenas a figura 5.11.

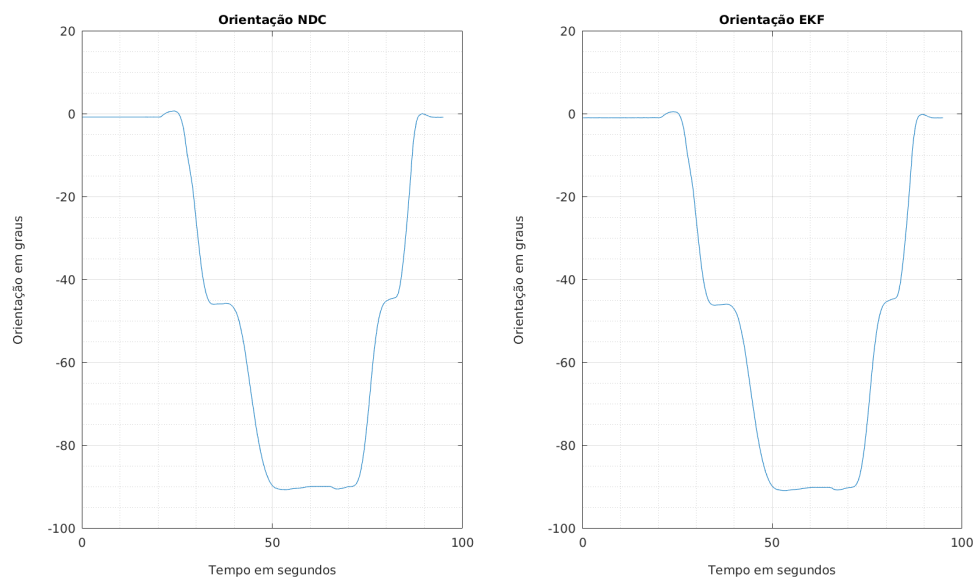


Figura 5.11: Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória

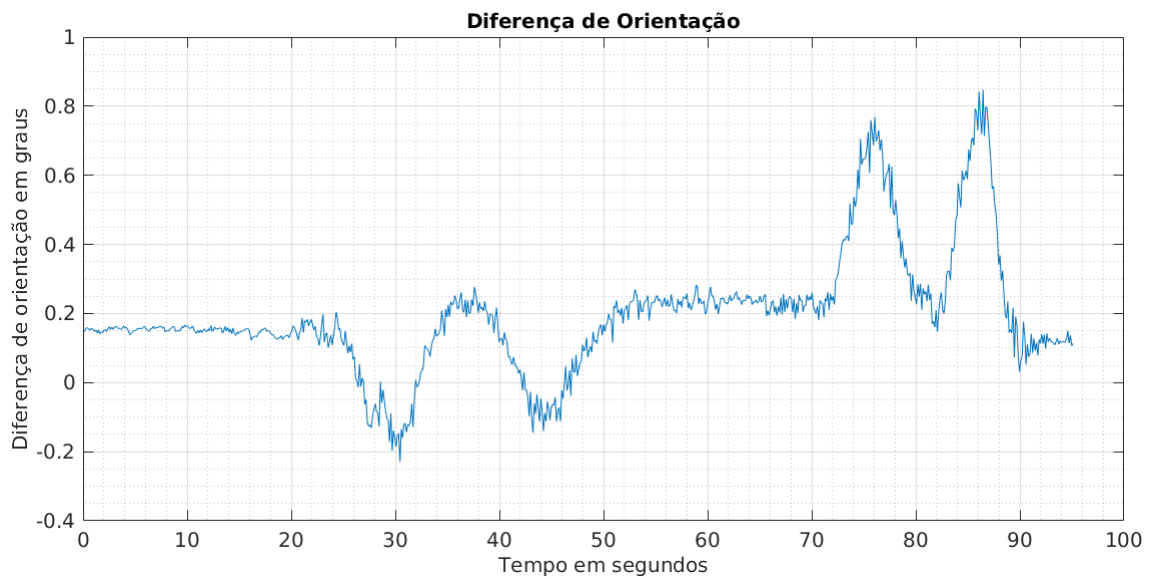


Figura 5.12: Diferença de orientação em graus entre orientação dada pelo NDC e EKF ao longo do teste

De forma semelhante ao que acontece na comparação entre o sistema da NDC e o laser *SICK NAV350*, existem quatro momentos onde o erro entre o sistema da NDC e o *Extended Kalman Filter*, que coincidem com os momentos onde o AGV executa as curvas 2 e 3 nos em $t = 30$ s, $t = 45$ s e $t = 76$ s, $t = 86$ s, pela primeira e segunda vez, respetivamente. Neste caso o erro máximo de orientação é também ligeiramente inferior, 0.8473° , comparativamente com os 1.0032° da comparação anterior. Além do erro máximo de orientação, também o erro médio e o desvio padrão são ligeiramente inferiores, 0.1954° e 0.1764° respetivamente, em comparação com 0.2432° e 0.1889° da comparação entre o NDC e o *SICK NAV350*.

É também visível na figura 5.12 um erro em torno de 0.2° . Uma vez que o *Extended Kalman Filter* utiliza as medidas do laser *SICK NAV350* era de esperar este desvio na orientação em relação ao sistema da NDC.

5.4 NAV350 vs EKF com detecção de refletores NAV350

Aqui será comparado o sistema de localização do próprio laser *SICK NAV350* com o *Extended Kalman Filter* que utiliza as medidas do laser *SICK NAV350*. Ao contrário do que acontecia na comparação entre o sistema da NDC com o sistema de localização do laser *SICK NAV350* e do sistema da NDC com *Extended Kalman Filter*, em que os resultados obtidos eram calculados a partir de medidas registadas em lasers diferentes, ou seja, tendo por base referenciais diferentes.

Neste caso os resultados obtidos terão por base as medidas do mesmo laser de navegação, o *SICK NAV350*, e consequentemente as medidas serão relativas ao mesmo referencial. Trata-se de uma vantagem, que na teoria levará a uma menor diferença entre os sistemas analisados. Em contraste a isto, como se verificou nas duas comparações anteriores, o facto de as medidas

Diferença absoluta média de posição	0.0172 m
Desvio padrão da diferença de posição	0.0068 m
Diferença máxima de posição	0.0385 m
Diferença média de orientação	0.1954 °
Desvio padrão da diferença de orientação	0.1764 °
Diferença máxima de orientação	0.8473 °

Tabela 5.2: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF

serem relativas a dois referenciais diferentes leva a existência de erros devidos à posição de um referencial relativamente ao outro não ser exata. Um desses erros foi o *offset* de 0.2° verificado nas duas comparações anteriores.

É possível verificar na figura 5.13 que a posição dada pelo sistema de localização do laser *SICK NAV350* tem algumas oscilações mais bruscas aquando da curva 3 relativamente à posição dada pelo *Extended Kalman Filter*. Tal é evidenciado na figura 5.14. Essas oscilações também são mais visíveis na figura 5.15. Em $t = 24$ s, onde o AGV faz a curva 2 pela primeira vez, a diferença de posição atinge o valor máximo de 0.0205 m.

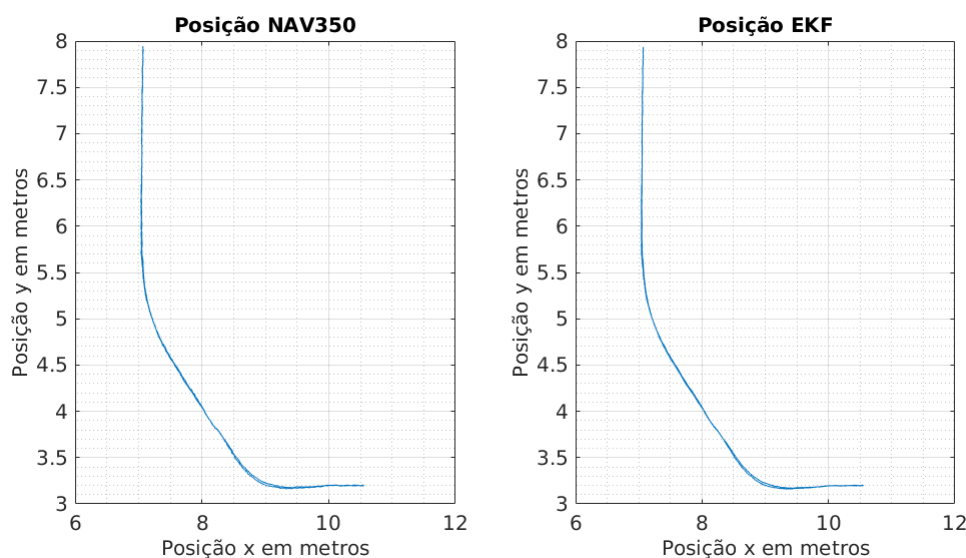


Figura 5.13: Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória

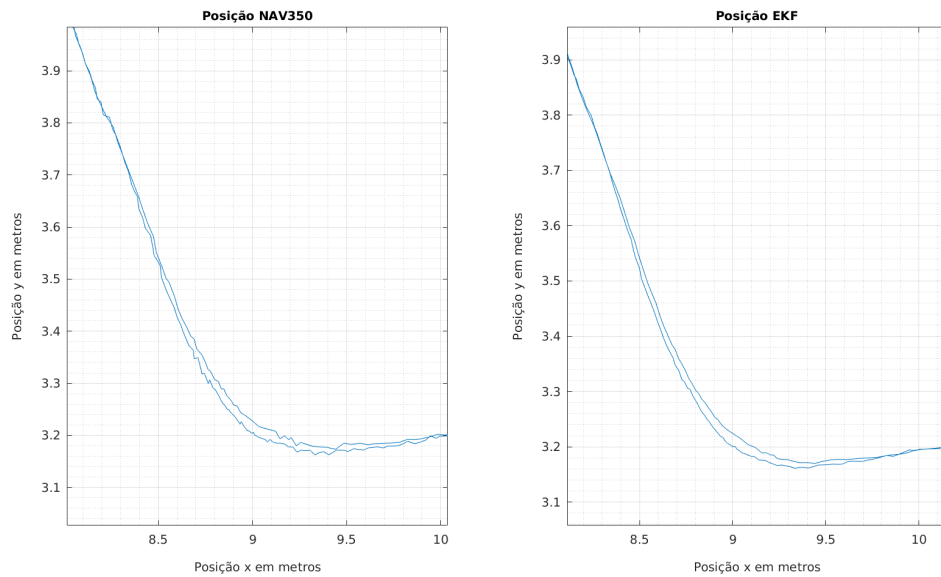


Figura 5.14: Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a curva 2 evidenciada

Como era esperado a diferença média é substancialmente mais baixa. Neste caso toma o valor de 0.0075 m, com desvio padrão de 0.0040 m, em comparação com 0.0169 m da comparação entre o NDC e o SICK NAV350 e 0.0172 m da comparação entre o NDC e o EKF.

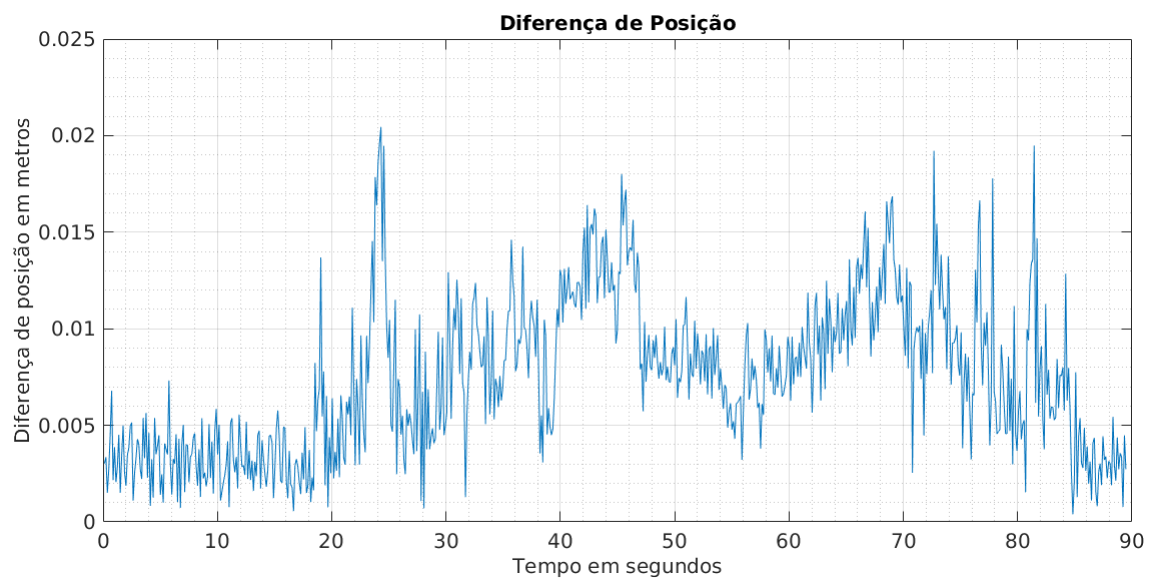


Figura 5.15: Diferença de posição em metros entre posição dada pelo NAV350 e EKF ao longo do teste

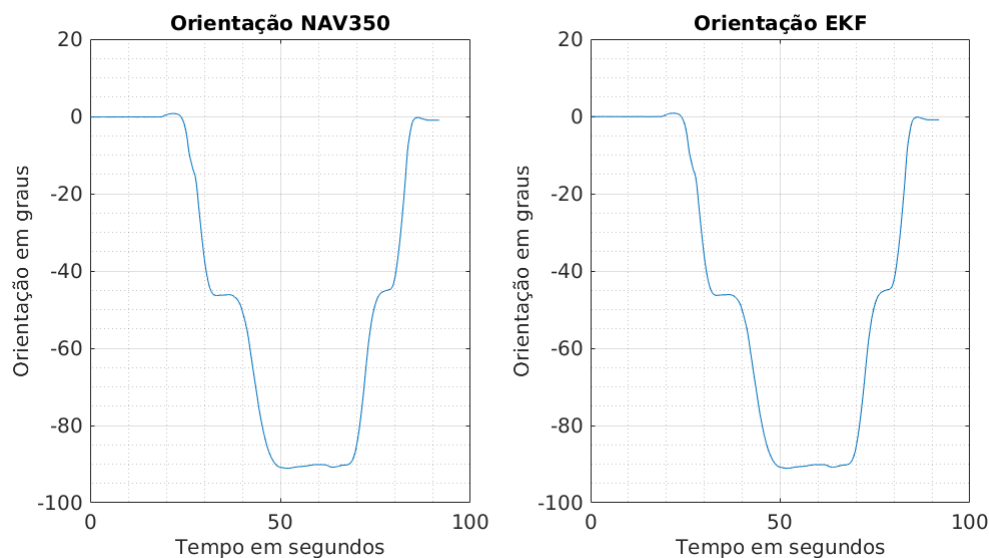


Figura 5.16: Na figura à esquerda está representada a orientação, em graus, dada pelo NAV350 ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória

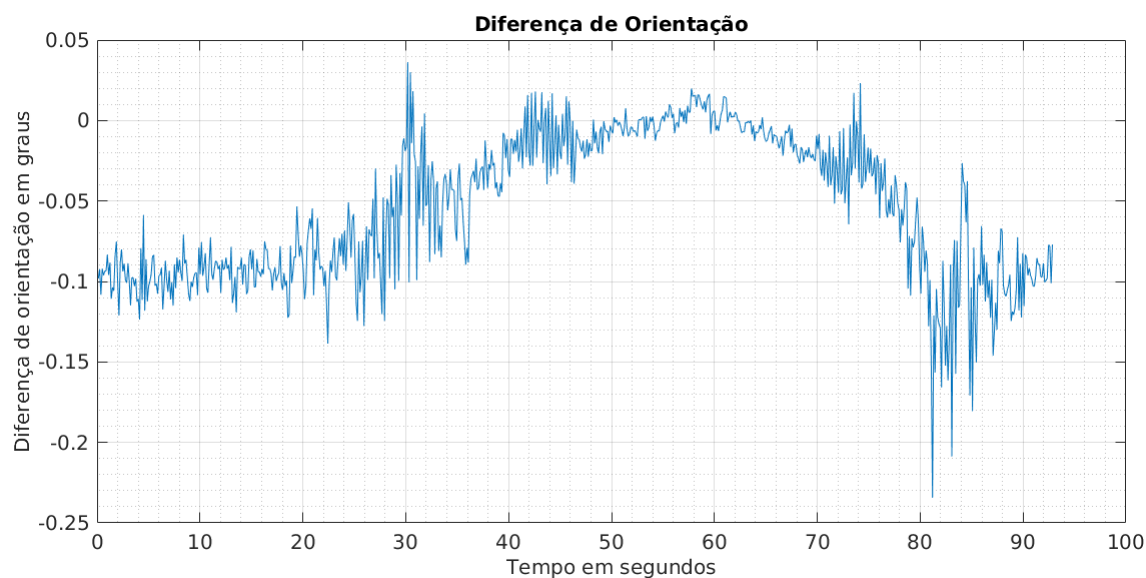


Figura 5.17: Diferença de orientação em graus entre orientação dada pelo NAV350 e EKF ao longo do teste

No que diz respeito à orientação dada entre os dois sistemas, como se esperava, a diferença média é substancialmente inferior, cerca de -0.0539° , comparativamente à diferença média registada nos testes entre o sistema da NDC com SICK NAV350, cerca de 0.2432° e nos testes entre o NDC e o EKF, com uma diferença média de orientação de cerca de 0.1854° .

A diferença máxima de orientação volta a acontecer nos momentos onde o AGV executa as curvas 2 e 3, como se verificou em todas os testes. Neste caso a diferença máxima de orientação

foi de cerca de 0.0362° , valor bastante abaixo dos testes entre sistemas que utilizam medidas de dois lasers distintos.

Diferença absoluta média de posição	0.0075 m
Desvio padrão da diferença de posição	0.0040 m
Diferença máxima de posição	0.0205 m
Diferença média de orientação	-0.0539°
Desvio padrão da diferença de orientação	0.0448°
Diferença máxima de orientação	0.0362°

Tabela 5.3: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF

5.5 Scan to beacons vs Detecção de Beacons NAV350

Nos dois testes anteriores, o nó *Extended Kalman Filter* estava a utilizar as posições dos refletores detetados calculadas pelo próprio laser *SICK NAV350*, no entanto, antes dessa funcionalidade estar disponível, foi desenvolvido em (Sobreira, 2017), um nó ROS que utiliza as medidas vindas do laser, calcula as posições dos refletores e filtra possíveis *outliers*.

Durante a configuração da localização de refletores do laser *SICK NAV350* verificou-se que a distância mínima para a deteção de refletores era de 0.5 m. Isto, pode ser uma desvantagem em relação ao sistema desenvolvido em (Sobreira, 2017), pois durante o trajeto de testes, existem pontos onde o AGV está a menos de 0.5 metros de um refletor. Por isso, é de todo o interesse comparar-se os dois sistemas de localização de refletores.

Foi feito um teste onde foi registado o número de refletores detetados pelo sistema de deteção de refletores desenvolvido em (Sobreira, 2017) e o número de refletores detetados pelo próprio laser *SICK NAV350*. Sabe-se *a priori* que o número de refletores instalado no local de testes é de sete refletores. Por isso, caso um sistema detete um número superior a sete, é certo que algumas da deteções são falsos positivos. Isto não quer dizer que se forem detetados sete ou menos possíveis refletores todos eles sejam de facto refletores.

Do teste feito obteve-se os dados que estão na figura 5.18. Apesar de o sistema de deteção de refletores do laser *SICK NAV350* ter a restrição de apenas detetar refletores a um raio mínimo de 0.5 m, demonstrou um comportamento mais robusto, consistente e com um número de refletores detetados nunca superior a sete. Contrariamente, o sistema proposto em 5.18 teve resultados menos satisfatórios, tendo em muitos momentos detetado um número de refletores superior a sete.

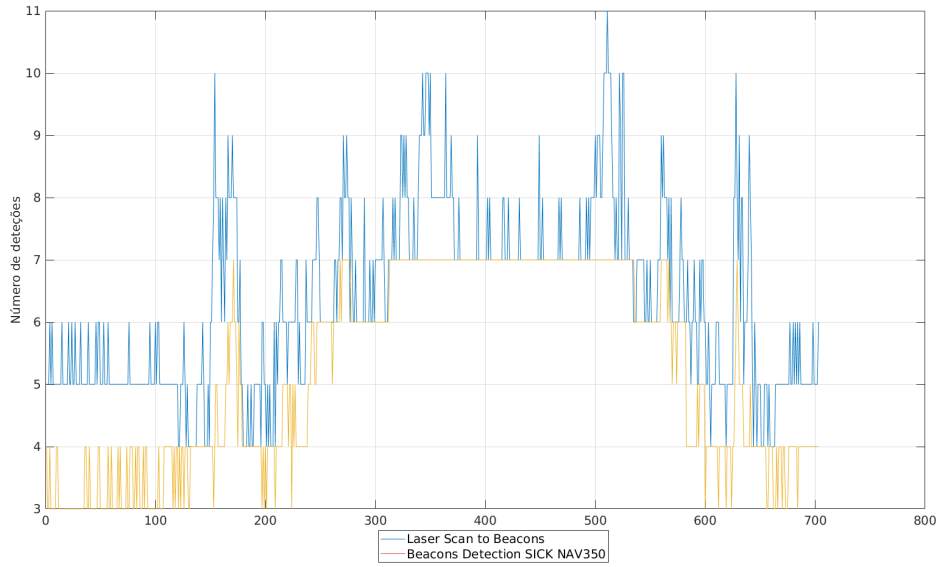


Figura 5.18: Número de possíveis refletores detetados pelos dois sistemas ao longo da trajetória pelo nó *Laser Scan to Beacons* e pelo próprio laser *SICK NAV350*

5.6 NDC vs EKF com detecção de refletores Laser Scan to Beacons

Como se identificaram diferenças significativas na detecção de refletores por parte do sistema desenvolvido em (Sobreira, 2017) em comparação com a detecção de refletores pelo laser *SICK NAV350*, decidiu-se testar o *Extended Kalman Filter* com a detecção de refletores proposta em (Sobreira, 2017), afim de verificar se essa diferença influenciava ou não nos resultados.

Ao analisar a figura 5.19, as diferenças não são perceptíveis, quer em relação à posição dada pelo sistema da NDC como em relação às posições dadas pelo *Extended Kalman Filter* do teste com a detecção de refletores feita pelo laser *SICK NAV350* (figura 5.8).

No entanto, ao analisar a figura 5.20 onde está representado a diferença de posição, é possível constatar que o padrão dessa diferença é bastante idêntica com a da comparação entre o NDC e o *Extended Kalman Filter* com os refletores detetados pelo laser *SICK NAV350*. Além disso, a diferença máxima ocorre quase no mesmo instante e toma o valor de 0.0423 m, ligeiramente superior a 0.0385 m, e a diferença média é de cerca de 0.0178 m com desvio padrão de 0.0070 m, também ligeiramente superior aos 0.0172 m com desvio padrão de 0.0068 m.

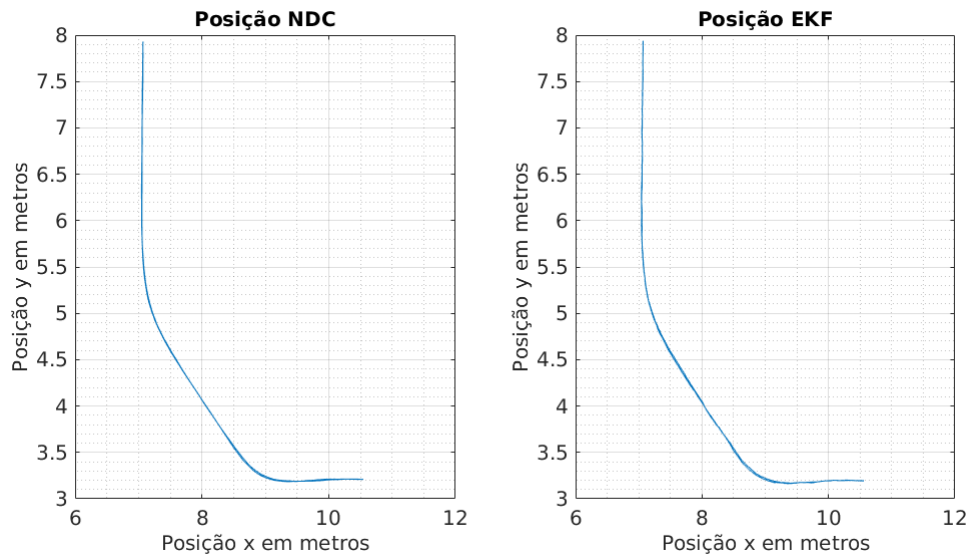


Figura 5.19: Na figura à esquerda está representada a posição dada pelo NDC ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó *Scan to Beacons*

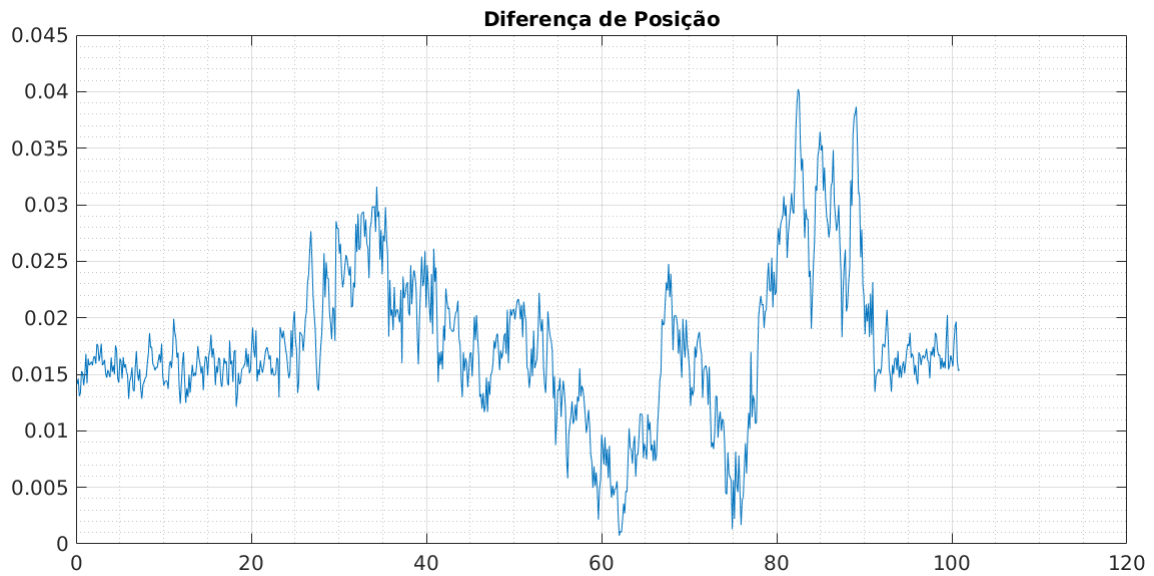


Figura 5.20: Diferença de posição em metros entre posição dada pelo NDC e EKF ao longo do teste, com a posição dos refletores dada pelo nó *Scan to Beacons*

No que diz respeito à orientação, o padrão apresentado pela diferença na figura 5.22 é bastante semelhante ao apresentado na figura 5.12. A diferença máxima de orientação é ligeiramente superior, ou seja, 0.9065° em comparação com 0.8473° . Apesar disso a diferença média de orientação foi inferior, ou seja, 0.1939° , com desvio padrão de 0.1788° em comparação com 0.1954° , com desvio padrão de 0.1764° .

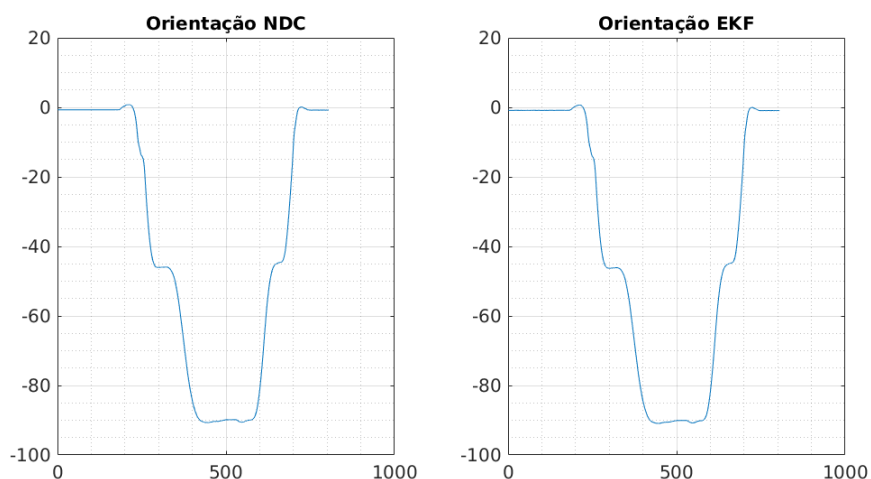


Figura 5.21: Na figura à esquerda está representada a orientação, em graus, dada pelo NDC ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó *Scan to Beacons*

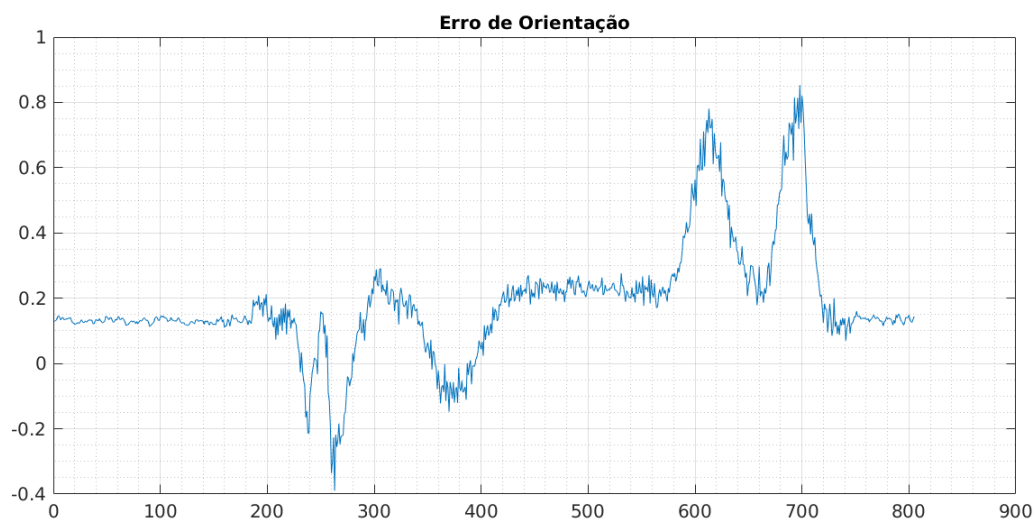


Figura 5.22: Diferença de orientação em graus entre orientação dada pelo NDC e EKF ao longo do teste, com a posição dos refletores dada pelo nó *Scan to Beacons*

Apesar do sistema de detecção de refletores apresentado em (Sobreira, 2017) apresentar resultados menos satisfatórios, a influencia disso não é notada na determinação da localização pelo *Extended Kalman Filter*, em comparação com os testes onde se utiliza a detecção de refletores do laser *SICK NAV350* no *Extended Kalman Filter*.

Diferença absoluta média de posição	0.0178 m
Desvio padrão da diferença de posição	0.0070 m
Diferença máxima de posição	0.0423 m
Diferença média de orientação	0.1939 °
Desvio padrão da diferença de orientação	0.1788 °
Diferença máxima de orientação	0.9065 °

Tabela 5.4: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF, com a posição dos refletores dada pelo nó *Scan to Beacons*

5.7 NAV350 vs EKF com detecção de refletores Laser Scan to Beacons

Uma vez que não se notaram diferenças significativas no teste entre a pose dada pelo sistema da NDC e o *Extended Kalman Filter* com a detecção de refletores proposto em (Sobreira, 2017), é de esperar que não se verifiquem diferenças significativas entre a comparação feita neste ponto e a comparação feita anteriormente entre a localização dada pelo laser *SICK NAV350* e o *Extended Kalman Filter* que usa a localização de refletores do próprio laser *SICK NAV350*.

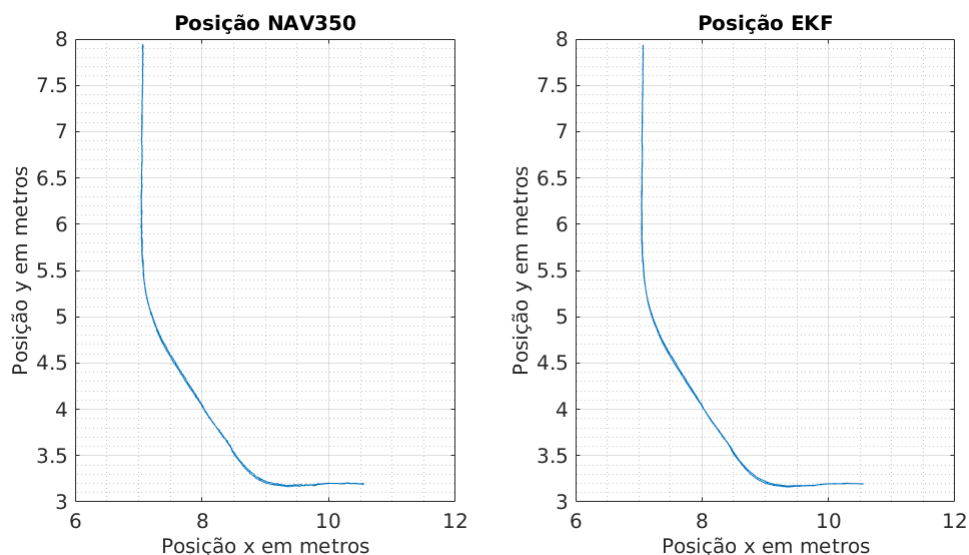


Figura 5.23: Na figura à esquerda está representada a posição dada pelo NAV350 ao longo da trajetória e à direita está representada a posição dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó *Scan to Beacons*

O padrão da diferença é idêntico ao da figura 5.15, sendo que nesta comparação, a diferença máxima de posição é ligeiramente superior, 0.0223 m, em comparação com 0.0205 m. No entanto, a diferença média de orientação nesta comparação é inferior, ou seja, 0.0074 m, com desvio padrão 0.0042 m, em contraste com 0.0075 m, com desvio padrão de 0.0040 m.

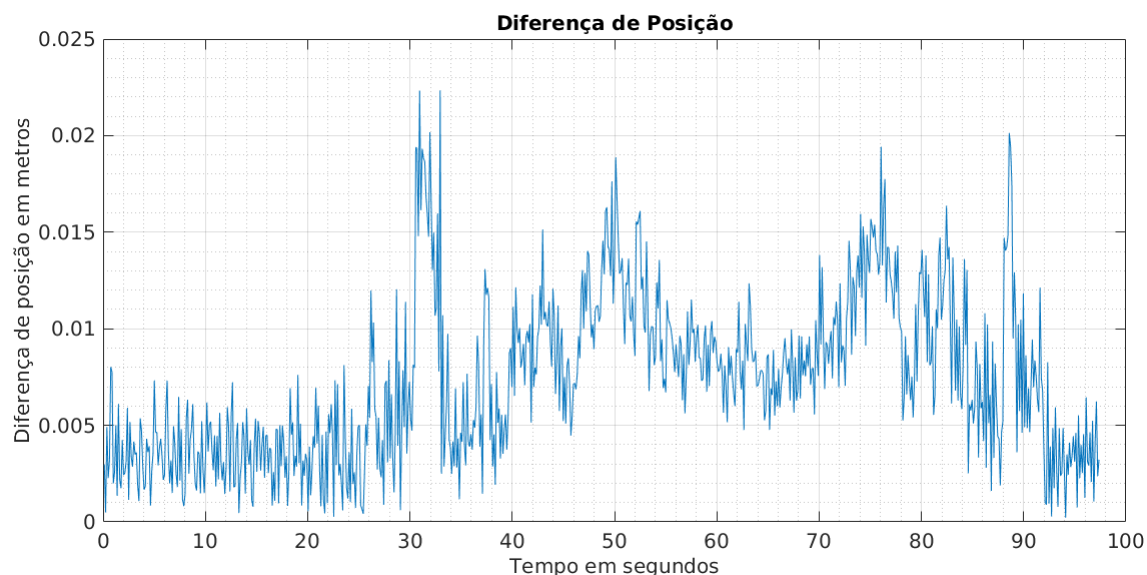


Figura 5.24: Diferença de posição em metros entre posição dada pelo NAV350 e EKF ao longo do teste, com a posição dos refletores dada pelo nó *Scan to Beacons*

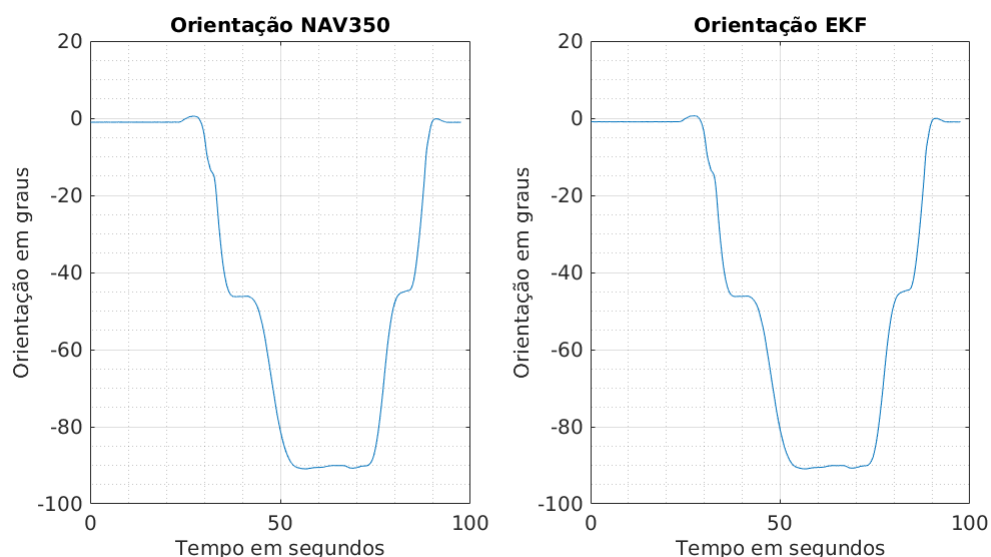


Figura 5.25: Na figura à esquerda está representada a orientação, em graus, dada pelo NAV350 ao longo da trajetória e à direita está representada a orientação dada pelo EKF ao longo da trajetória, com a posição dos refletores dada pelo nó *Scan to Beacons*

Diferença absoluta média de posição	0.0074 m
Desvio padrão da diferença de posição	0.0042 m
Diferença máxima de posição	0.0223 m
Diferença média de orientação	-0.0589 °
Desvio padrão da diferença de orientação	0.0465 °
Diferença máxima de orientação	0.0256 °

Tabela 5.5: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NAV350 e EKF, com a posição dos refletores dada pelo nó *Scan to Beacons*

No que diz respeito à orientação a diferença máxima é ligeiramente inferior, 0.0256° em relação à registada no teste com a detecção de refletores pelo laser *SICK NAV350*, de cerca de 0.0362° . Em relação à diferença média de orientação, foi registado aqui uma diferença média superior, ou seja, -0.0589° , com desvio padrão de 0.0465° em relação a -0.539° com desvio padrão de 0.0448° .

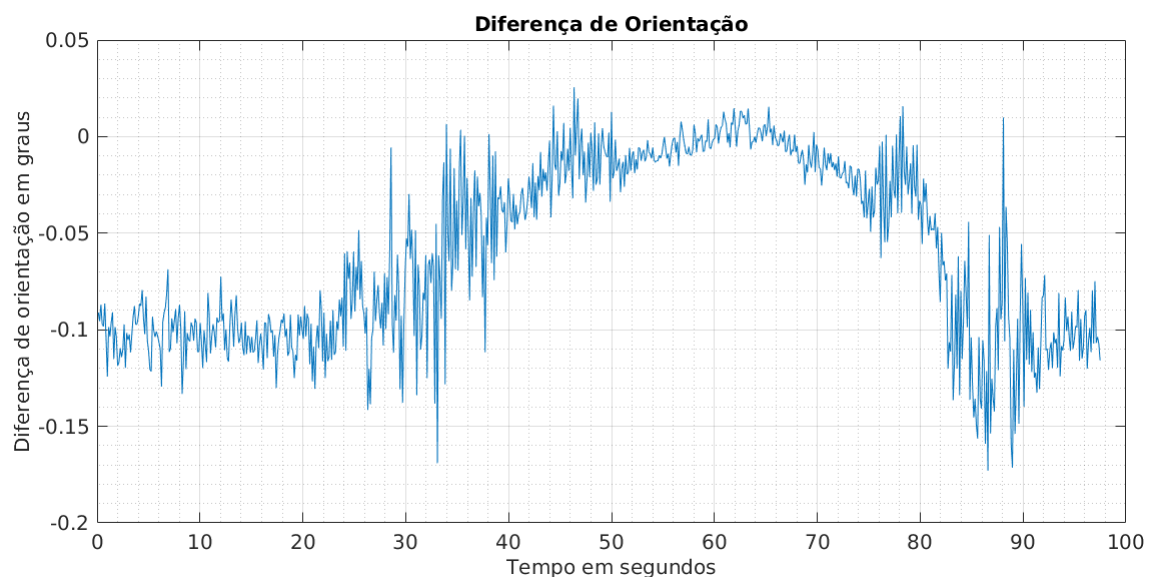


Figura 5.26: Diferença de orientação em graus entre orientação dada pelo NAV350 e EKF ao longo do teste, com a posição dos refletores dada pelo nó *Scan to Beacons*

5.8 Conclusões

Em suma, os sistemas que apresentaram diferenças menores entre si, tanto a nível da posição como da orientação foram o sistema de localização do laser *SICK NAV350* e o *Extended Kalman Filter*.

Esta diferença acentuada deve-se ao facto de nas comparações entre "NDC vs SICK NAV350" e "NDC vs EKF" estarem a ser utilizados dados de sensores que não estão perfeitamente alinhados. Isto poderá ser futuramente ser ajustado e calibrado, o que levará, na teoria, a uma diminuição das diferenças. Nas duas comparações onde se usou o *Extended Kalman Filter*, usou-se a detecção de refletores por parte do próprio laser *SICK NAV350*.

Quando se testou o "*Extended Kalman Filter*" com a detecção de refletores feita pelo nó *Laser Scan to Beacons* as diferenças não foram significativas comparativamente com os dados obtidos nos testes onde se usava a detecção de refletores feita pelo próprio laser *SICK NAV350*.

Como se verifica na tabela 5.7 não existe grande diferença na posição dada pelo *Extended Kalman Filter*, entre o uso da detecção de refletores do laser *SICK NAV350* ou o uso de detecção de refletores do *Laser Scan to Beacons*.

	NDC vs SICK NAV350	NDC vs EKF	SICK NAV350 vs EKF
Diferença absoluta média de posição	0.0169 m	0.0172 m	0.0075 m
Desvio Padrão da Diferença de Posição	0.0083 m	0.0068 m	0.0040 m
Diferença Máxima de Posição	0.0441 m	0.0385 m	0.0205 m
Diferença Média de Orientação	0.2432 °	0.1954 °	-0.0539 °
Desvio Padrão da Diferença de Orientação	0.1889 °	0.1764 °	0.0448°
Diferença Máxima de Orientação	1.0032 °	0.8473°	0.0362°

Tabela 5.6: Tabela com os valores calculados a partir dos dados das três comparações

É importante ter em conta o *offset* devido ao desalinhamento entre o laser usado pelo sistema da NDC e os sistemas que usam o laser *SICK NAV 350* de cerca de 0.2 °. Ou seja, A diferença média de orientação entre o sistema da NDC e a localização do *SICK NAV350* e a diferença média de orientação entre o sistema da NDC e o *Extended Kalman Filter* na realidade será inferior à registada.

	NDC vs EKF	NDC vs EKF (LStB)	NAV350 vs EKF	NAV350 vs EKF (LStB)
Diferença absoluta média de posição	0.0172 m	0.0178 m	0.0075 m	0.0074 m
Desvio Padrão da Diferença de Posição	0.0068 m	0.0070 m	0.0040 m	0.0042 m
Diferença Máxima de Posição	0.0385 m	0.0423 m	0.0205 m	0.0223 m
Diferença Média de Orientação	0.1954 °	0.1939 °	-0.0539 °	-0.0589 °
Desvio Padrão da Diferença de Orientação	0.1764 °	0.1788 °	0.0448 °	0.0465 °
Diferença Máxima de Orientação	0.8473 °	0.9065 °	0.0362 °	0.0256 °

Tabela 5.7: Comparação entre o *Extended Kalman Filter* com o uso de detecção de refletores do SICK NAV350, e com o uso de detecção de refletores com *Laser Scan to Beacons* (LSbT) e os restantes algoritmos

5.9 Comparações entre NDC, *SICK NAV350* e *Extended Kalman Filter* com o *Perfect Match*

Aqui serão comparados os três sistemas de localização com base em refletores, NDC, *SICK NAV350* e *Extended Kalman Filter* com o algoritmo de localização baseado em contornos *Perfect Match* que no que diz respeito ao estado da arte, já foi largamente analisado, contudo ainda não é muito utilizado comercialmente. É feita esta comparação para perceber como se comporta este algoritmo em relação aos outros algoritmos já comercializados.

Nesta fase do desenvolvimento, parte da trajetória até então utilizada nos testes ficou obstruída. Devido a isso, foi necessário utilizar apenas parte da trajetória para a realização dos testes. Como se pode ver na figura 5.27 a trajetória de testes ficou limitada ao percurso entre o ponto 5 e o ponto 3, sendo que apenas pode ser executado nesse sentido. No início do projeto, como já foi referido os responsáveis da *CONSOVEYO* definiram uma trajetória, que apenas pode ser alterada pelos mesmos, que permite o robô ir de modo autónomo desde o ponto 1 ao ponto 5 e vice-versa. Para ultrapassar isto, o AGV foi colocado de forma manual no ponto 5 e é dada a ordem para ir de forma autónoma até ao ponto 1 e parado manualmente quando o mesmo chega ao ponto 3, após isto é colocado de novo no ponto 5 de forma manual.

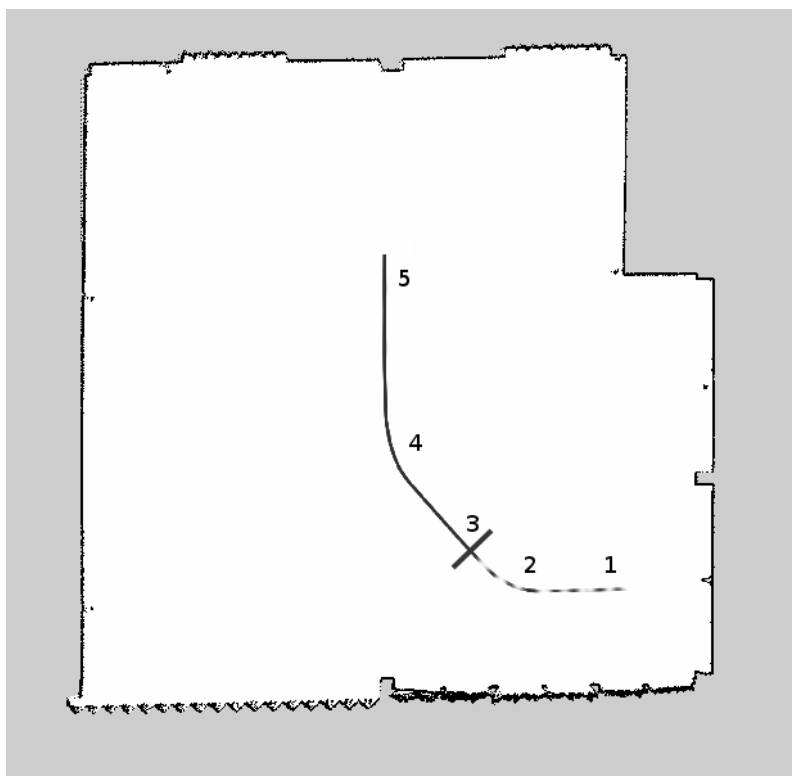


Figura 5.27: Mapa do espaço feito a partir do *package* ROS Hector Mapping com a trajetória de testes que o AGV efetuou nas comparações entre o *Perfect Match* e os restantes algoritmos

5.9.1 Perfect Match vs NDC

Como se verifica na figura 5.28 a posição dada pelo sistema da NDC é bastante mais suave que a trajetória calculada pelo algoritmo de localização por contornos *Perfect Match* nas partes onde a trajetória é curvilínea, no entanto, nas partes onde a trajetória é retilínea ambos os algoritmos apresentam uma trajetória bastante suave e idêntica. O erro absoluto médio de posição foi de 0.0294 m, (tabela 5.8), este valor foi ligeiramente superior aos valores obtidos quando se comparou o sistema da NDC com o *Extended Kalman Filter*, que se obteve 0.0172 m e com o *SICK NAV 350*, que se obteve 0.0169 m.

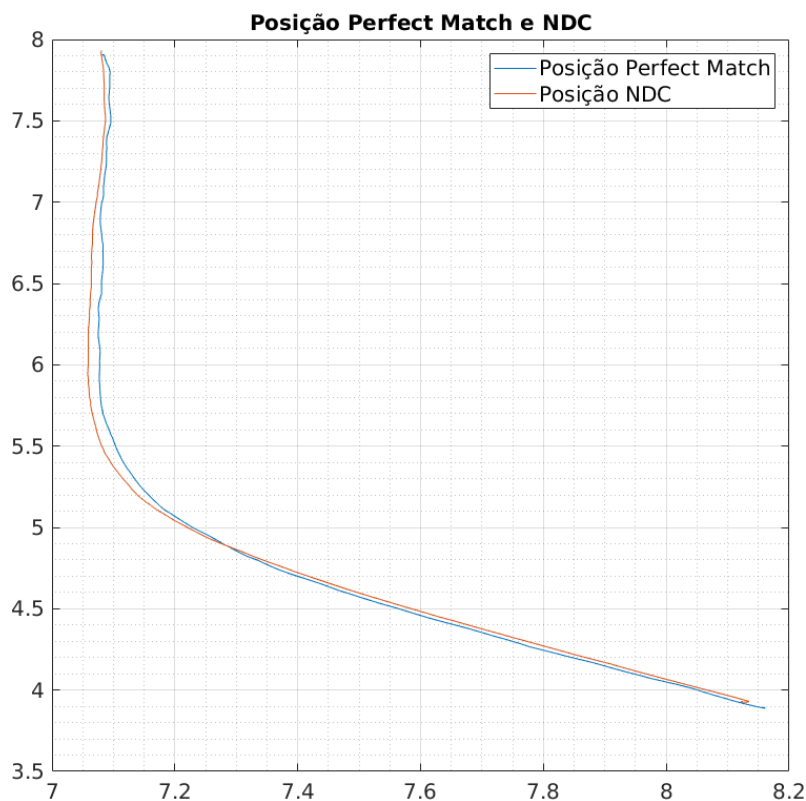


Figura 5.28: A azul está representada a posição dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a posição do sistema de localização por refletores da NDC durante a execução da trajetória

No que diz respeito à diferença de posição é possível verificar o mesmo padrão observado que aconteceu em praticamente todos os testes, de a diferença aumentar nos locais onde a trajetória é curvilínea. Esses momentos aconteceram, como se pode ver na figura 5.29, entre o segundo 10 e segundo 20 onde o robô efetua um pequeno ajuste para se enquadrar com a trajetória e a partir do segundo 25 onde o robô executa a curva 4 (figura 5.27).

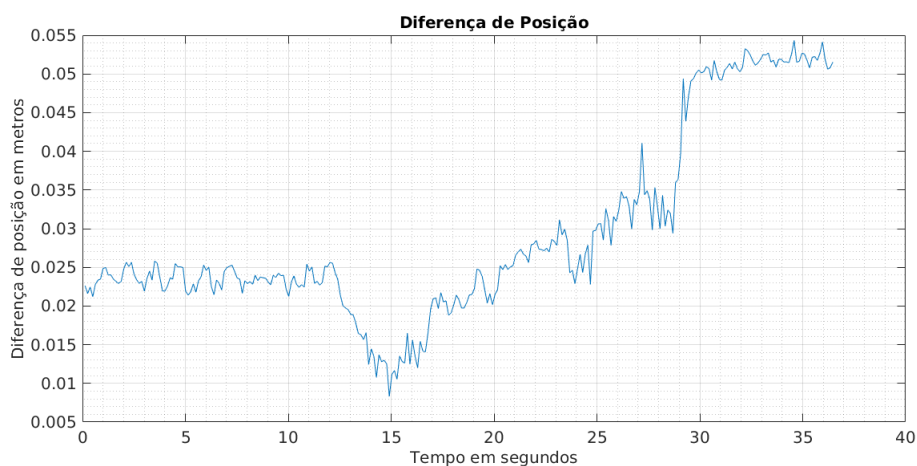


Figura 5.29: Diferença absoluta entre a posição dada pelo sistema da NDC e o algoritmo de localização com base em contornos *Perfect Match*

Já em relação à orientação é possível ver um *offset* de cerca de 0.2° ao longo de toda a trajetória. Como já tinha sido notado anteriormente, isto deve-se ao facto de o laser que o sistema da NDC utiliza para aquisição de medidas e o laser *SICK NAV350* que o *Perfect Match* utiliza estão desalinhados em cerca de 0.2° , daí este *offset* aparecer em todas as comparações entre o sistema da NDC e os restantes que utilizam o laser *SICK NAV 350*.

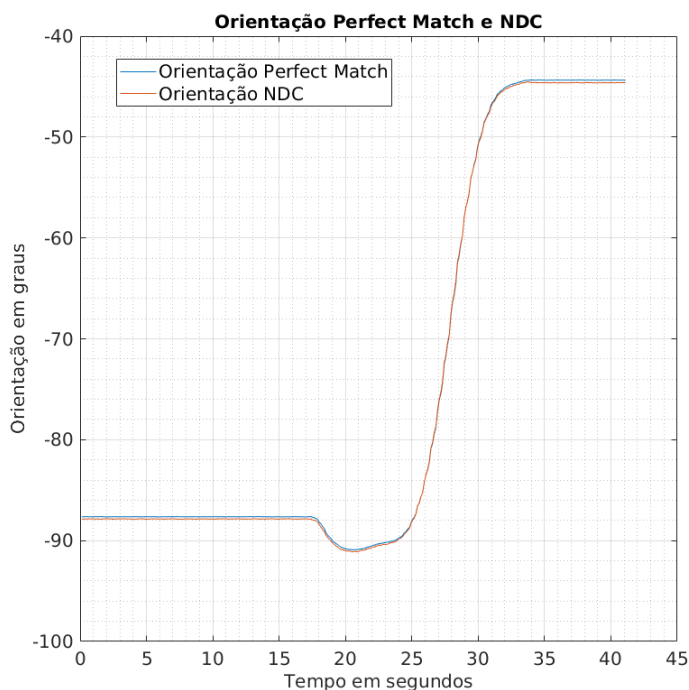


Figura 5.30: A azul está representada a orientação dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a orientação do sistema de localização por refletores da NDC durante a execução da trajetória

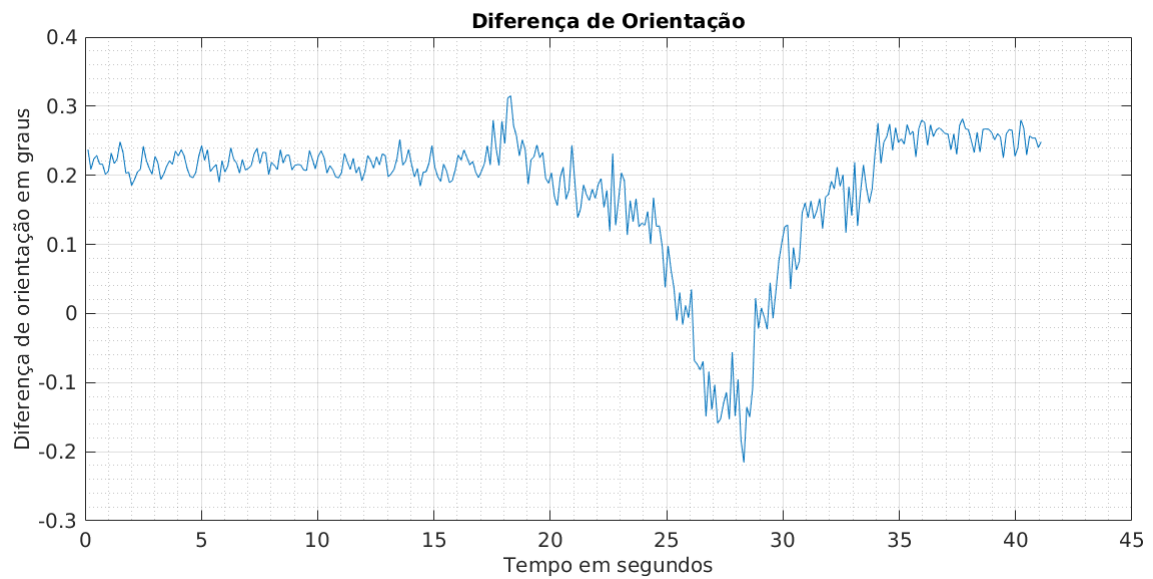


Figura 5.31: Diferença de orientação entre a posição dada pelo sistema da NDC e o algoritmo de localização com base em contornos *Perfect Match*

Este *offset* de 0.2° é facilmente observável na figura 5.31, onde também se está presente um aumento da diferença entre as orientações no momento em que o robô executa a curva 4.

Diferença absoluta média de posição	0.0294 m
Desvio padrão da diferença de posição	0.0121 m
Diferença máxima de posição	0.0543 m
Diferença média de orientação	0.1790°
Desvio padrão da diferença de orientação	0.0997°
Diferença máxima de orientação	0.3148°

Tabela 5.8: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o *Perfect Match* e o NDC

5.9.2 *Perfect Match* vs NAV350

De testes anteriores verificamos que a posição dada pelo laser *SICK NAV 350* é um pouco menos suave quando comparada com o sistema da NDC, ou o *Extended Kalman Filter*, tal acontece também aqui. No entanto, o *Perfect Match* é ainda menos suave que o *SICK NAV 350* em certos pontos da trajetória. Já na parte final da trajetória tanto o *Perfect Match* como a localização do laser *SICK NAV 350* são bastante suaves e idênticas, como se pode verificar na figura 5.32.

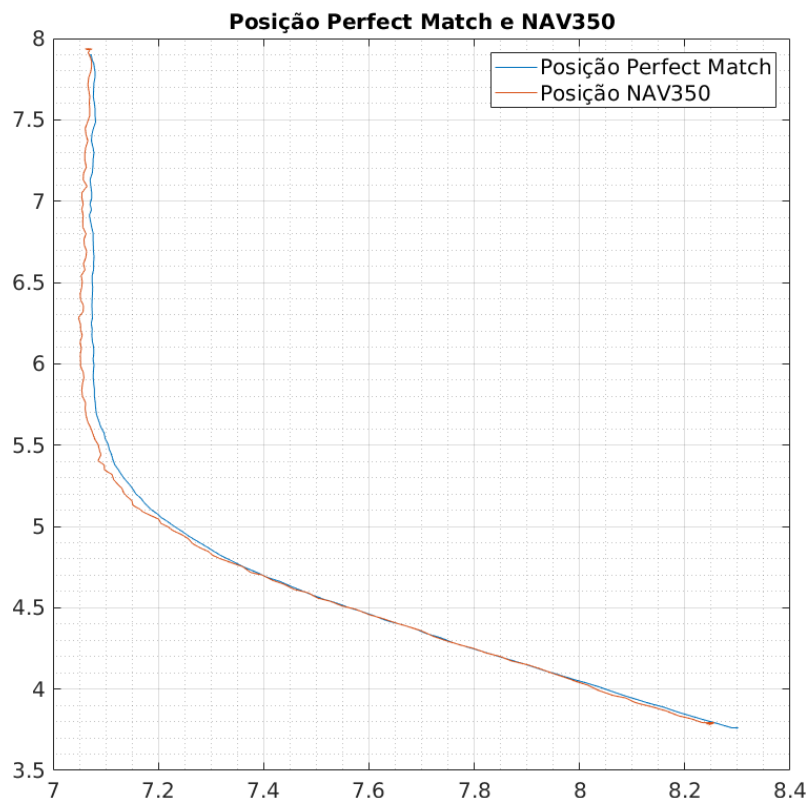


Figura 5.32: A azul está representada a posição dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a posição do sistema de localização por refletores do laser *SICK NAV350* durante a execução da trajetória

Mais uma vez, os resultados referentes à diferença absoluta de posição são idênticos aos obtidos em comparações com a localização do *SICK NAV350*. O erro tende a ser constante no segmentos da trajetória retilínea e aumentam quando o robô executa as curvas, ou seja, entre o segundo 15 e o segundo 25, como se pode verificar na figura 5.33. Ainda assim, quando comparada a localização do *SICK NAV350* com o sistema da NDC, onde se obteve um erro absoluto médio de posição de 0.0169 m, ou se comparou o *SICK NAV350* com o *Extended Kalman Filter*, onde se obteve um erro absoluto médio de posição de 0.0075 m, ao comparar a localização do laser *SICK NAV350* com o *Perfect Match* obteve-se uma diferença absoluta média superior de cerca de 0.0452 m.

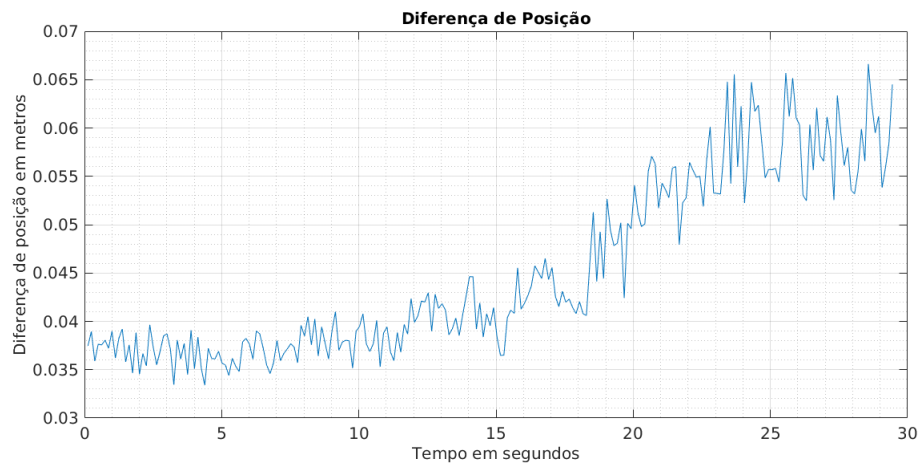


Figura 5.33: Diferença absoluta entre a posição dada pelo próprio laser *SICK NAV350* e o algoritmo de localização com base em contornos *Perfect Match*

Relativamente a diferença de orientação entre o *Perfect Match* e o *SICK NAV350* é notório um *offset* entre ambas. Neste caso o *offset* não é devido ao desalinhamento entre *lasers* pois o *Perfect Match* utiliza os dados vindos do próprio laser *SICK NAV350*. Neste caso o *offset* deve-se ao facto de alguns dos parâmetros do *Perfect Match* não estarem precisamente calibrados. Como se pode ver na figura 5.35 esse *offset* é de cerca de 0.46° .

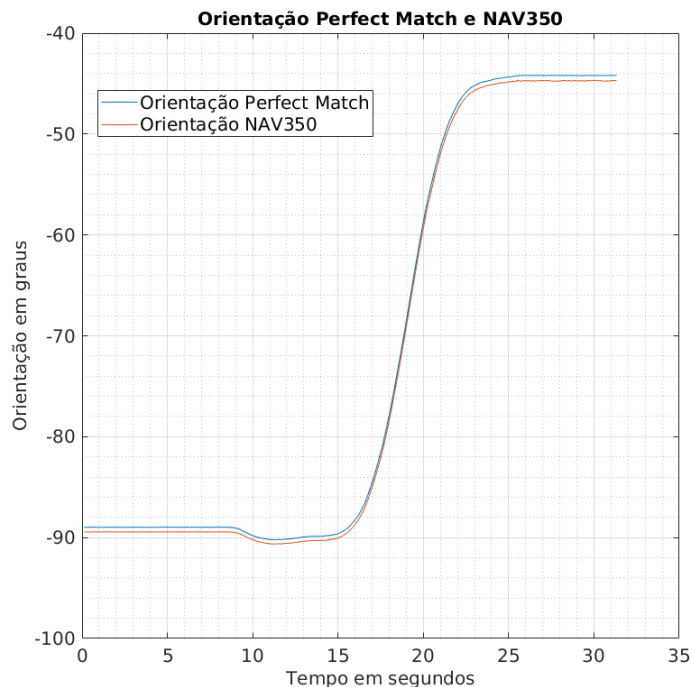


Figura 5.34: A azul está representada a orientação dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a orientação do sistema de localização por refletores do laser *SICK NAV350* durante a execução da trajetória

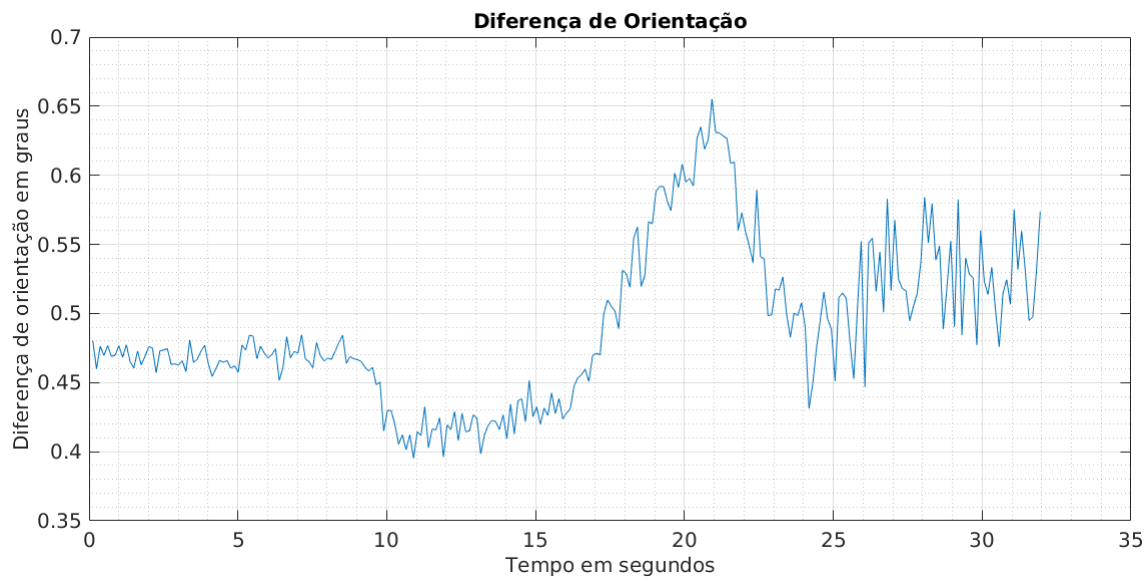


Figura 5.35: Diferença de orientação entre a orientação dada pelo laser *SICK NAV350* e o algoritmo de localização com base em contornos *Perfect Match*

Como se pode ver na tabela 5.9 a diferença média de orientação é relativamente alta, cerca de 0.4907° , no entanto tal valor está afetado com *offset* de cerca de 0.46° , o que torna esse valor um pouco enganador.

Diferença absoluta média de posição	0.0452 m
Desvio padrão da diferença de posição	0.0091 m
Diferença máxima de posição	0.0666 m
Diferença média de orientação	0.4907°
Desvio padrão da diferença de orientação	0.0569°
Diferença máxima de orientação	0.6552°

Tabela 5.9: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e *SICK NAV350*

5.9.3 Perfect Match vs EKF

É de esperar nesta comparação que os resultados sejam muito idênticos aos obtidos entre o *Perfect Match* e o *Extended Kalman Filter*, pois de comparações entre a localização do laser *SICK NAV350* e o *Extended Kalman Filter* obtiveram-se resultados muito idênticos entre estes dois sistemas de localização, sendo que o *Extended Kalman Filter* é um pouco mais suave.

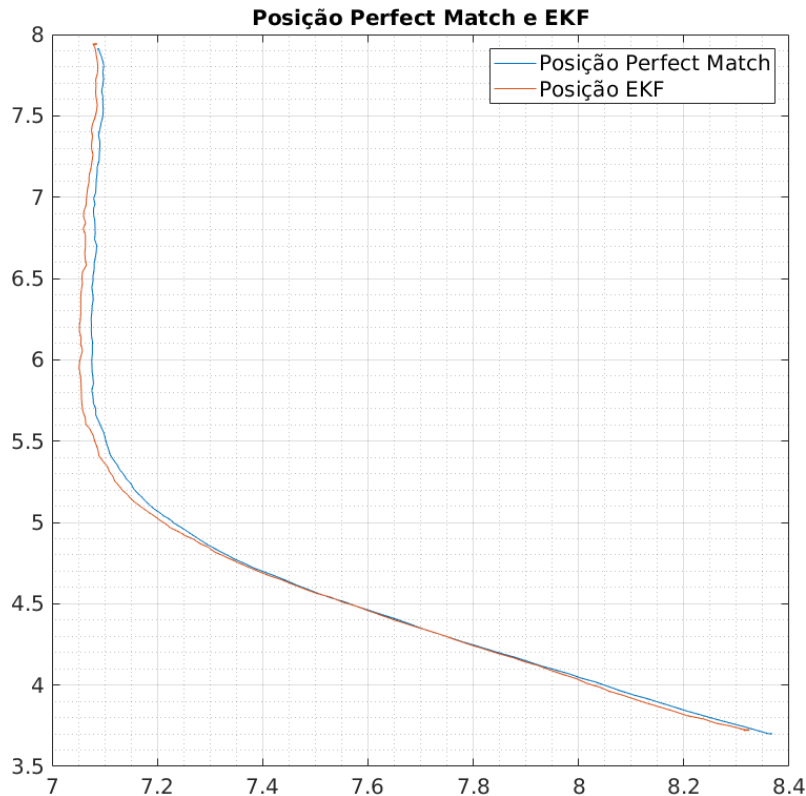


Figura 5.36: A azul está representada a posição dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a posição do algoritmo *Extended Kalman Filter*, durante a execução da trajetória

Como era de esperar, o *Extended Kalman Filter* apresenta uma trajetória quando comparado com o *SICK NAV350* e consequentemente com o *Perfect Match*, como se pode ver na figura 5.36. Ao analisar a figura 5.37 é visível o padrão observado na figura 5.33, como era esperado. O erro médio absoluto de posição é constante durante a parte da trajetória retilínea e aumenta no momento onde o robô executa a curva 4.

No que diz respeito à orientação, o *offset* observado na figura 5.35 da comparação anterior, é também visível nesta comparação na figura 5.39 o *offset* de cerca de 0.46° , detetado na comparação entre o *Perfect Match* e a localização por parte do laser *SICK NAV350*.

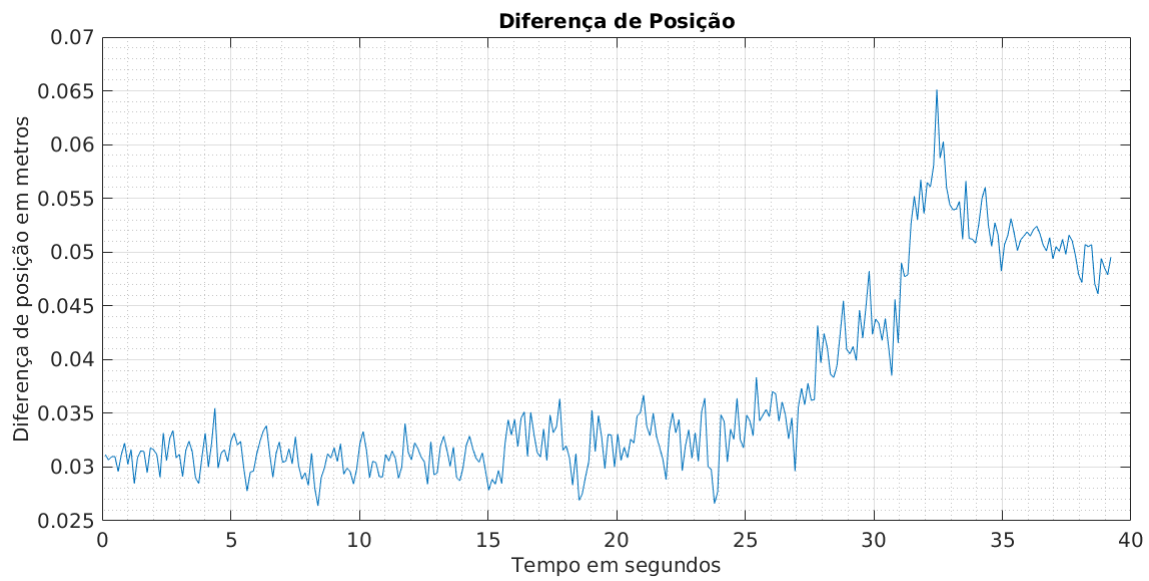


Figura 5.37: Diferença absoluta entre a posição dada pelo algoritmo *Extended Kalman Filter* e o algoritmo de localização com base em contornos *Perfect Match*

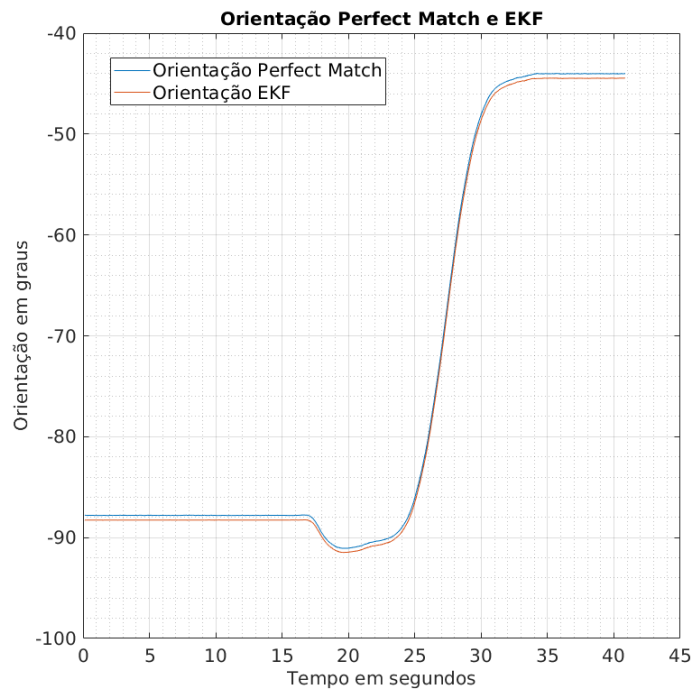


Figura 5.38: A azul está representada a orientação dada pelo algoritmo de localização com base em contornos *Perfect Match* e a vermelho a orientação do algoritmo *Extended Kalman Filter* durante a execução da trajetória

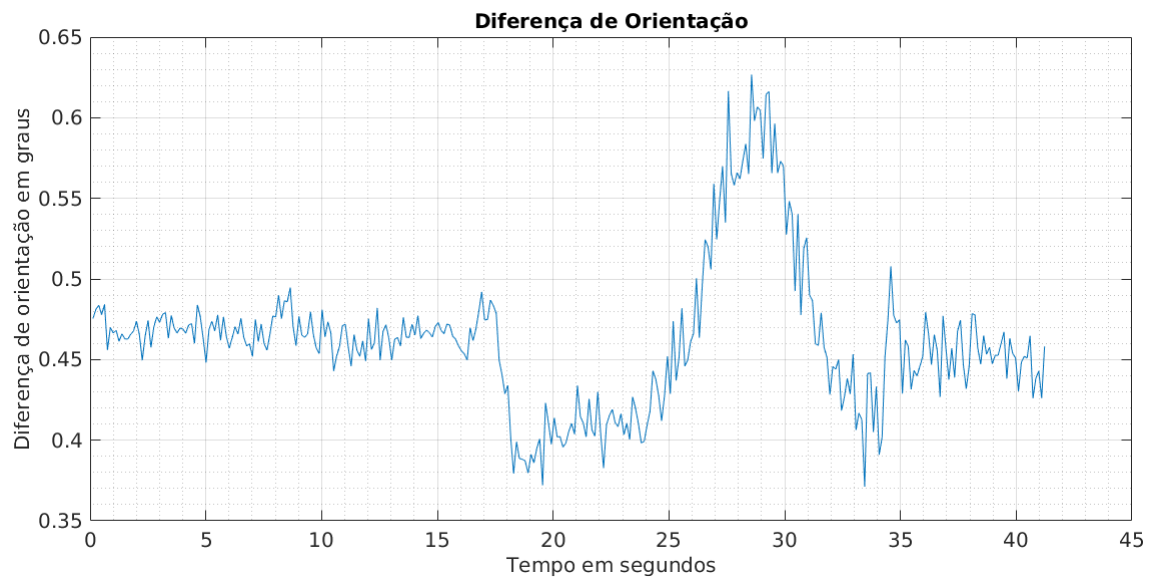


Figura 5.39: Diferença de orientação entre do algoritmo *Extended Kalman Filter* e o algoritmo de localização com base em contornos *Perfect Match*

Diferença absoluta média de posição	0.0369 m
Desvio padrão da diferença de posição	0.0087 m
Diferença máxima de posição	0.0651 m
Diferença média de orientação	0.4624 °
Desvio padrão da diferença de orientação	0.0445 °
Diferença máxima de orientação	0.6267 °

Tabela 5.10: Diferença média, desvio padrão e diferença máxima da posição e orientação do teste entre o NDC e EKF, com a posição dos refletores dada pelo nó *Scan to Beacons*

5.9.4 Conclusões

Da comparação entre o algoritmo de localização baseada em contornos *Perfect Match* e os três algoritmos de localização com base em refletores, sistema da NDC, localização do laser *SICK NAV350* e o algoritmo *Extended Kalman Filter* verificou-se que o *Perfect Match* é um pouco mais instável e menos suave que os outros três métodos, no entanto as diferenças de posição e orientação não foram muito acentuadas.

É de salientar que dois dos sistemas utilizados (NDC e *SICK NAV350*) para a comparação com o *Perfect Match*, são sistemas comerciais já largamente utilizados e validados e as diferenças entre o *Perfect Match* e esses dois sistemas não foram relativamente baixas.

Além disso o algoritmo de localização com base em contornos *Perfect Match* não necessita da instalação de *hardware* no meio onde o AGV navega, como por exemplo refletores, pois calcula a pose do robô tendo em conta apenas as características do meio. Este facto diminui bastante os custos do sistema pois não é necessária a instalação de refletores, que pode ser bastante dispendioso.

	Perfect Match vs NDC	Perfect Match vs NAV350	Perfect Match vs EKF
Diferença absoluta média de posição	0.0294 m	0.0452 m	0.0369 m
Desvio padrão da diferença de posição	0.0121 m	0.0091 m	0.0087 m
Diferença máxima de posição	0.0543 m	0.0666 m	0.0651 m
Diferença média de orientação	0.1790 °	0.4907 °	0.4624 °
Desvio padrão da diferença de orientação	0.0997 °	0.0569 °	0.0445 °
Diferença máxima de orientação	0.3148 °	0.6552 °	0.6267 °

Tabela 5.11: Comparação entre o *Perfect Match* com o *Extended Kalman Filter*, *SICK NAV350* e o sistema da NDC

Como se verifica na tabela 5.11 o algoritmo de localização por contornos *Perfect Match* apresenta resultados muito idênticos ao sistema comercial da NDC que utiliza refletores. De realçar que a diferença média absoluta de posição é de apenas 2 cm enquanto que a diferença média de orientação é de apenas cerca de 0.17 °.

5.10 Algoritmo de Localização Global

Para o teste do algoritmo implementado usou-se o *script MATLAB* onde este algoritmo foi inicialmente implementado e testado com dados fictícios. No entanto, neste ponto e para validação dos resultados obtidos com dados fictícios, utilizou-se o mapa de refletores usado no *Extended Kalman Filter* e na localização do laser *SICK NAV350*, assim como as posições do refletores detetados pelo laser *SICK NAV350*.

Com o AGV parado na sua posição inicial, ou seja, no ponto 1 da figura 5.2, foram registadas as posições dos refletores detetados pelo laser *SICK NAV350*, com essas posições e com o algoritmo proposto, foi calculada a estimativa da posição inicial do AGV.

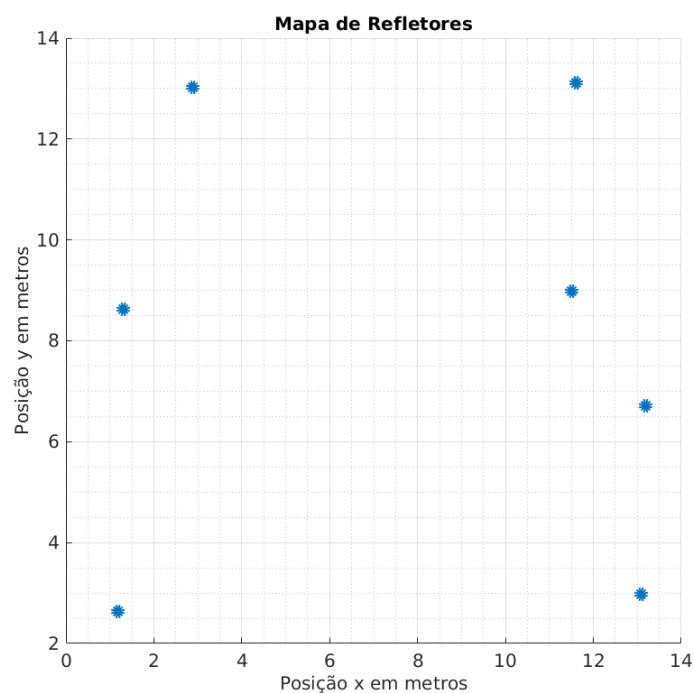


Figura 5.40: Mapa de refletores usado no algoritmo de localização global

Como se verifica na figura 5.41, foram detetados três refletores pelo laser *SICK NAV350*, representados no referencial do laser. Este é o número mínimo para que este algoritmo possa calcular uma estimativa da posição global do AGV.

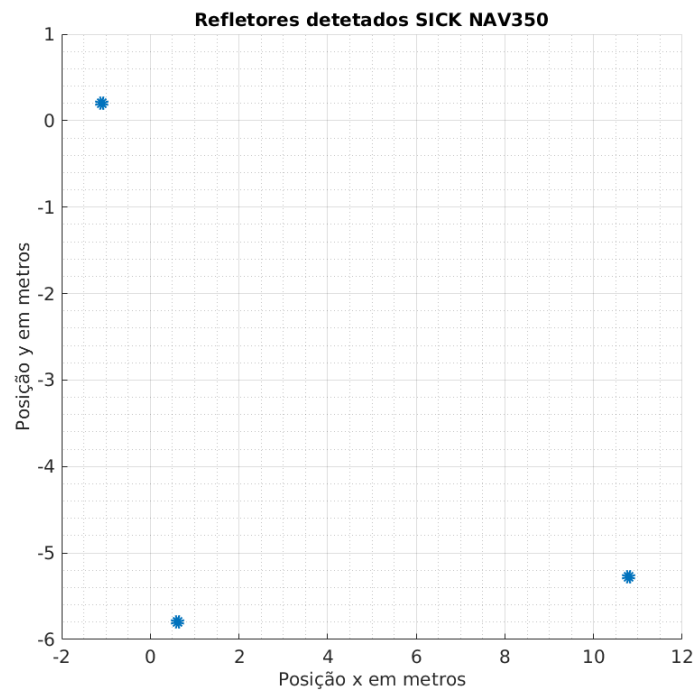


Figura 5.41: Refletores detetados pelo laser *SICK NAV350*

Com o número mínimo de observações o algoritmo convergiu para a solução correta como se pode verificar na figura 5.42 com uma diferença absoluta de posição de 0.03 metros e um uma diferença de ângulo de 0.08° em relação à pose dada pelo sistema da NDC.

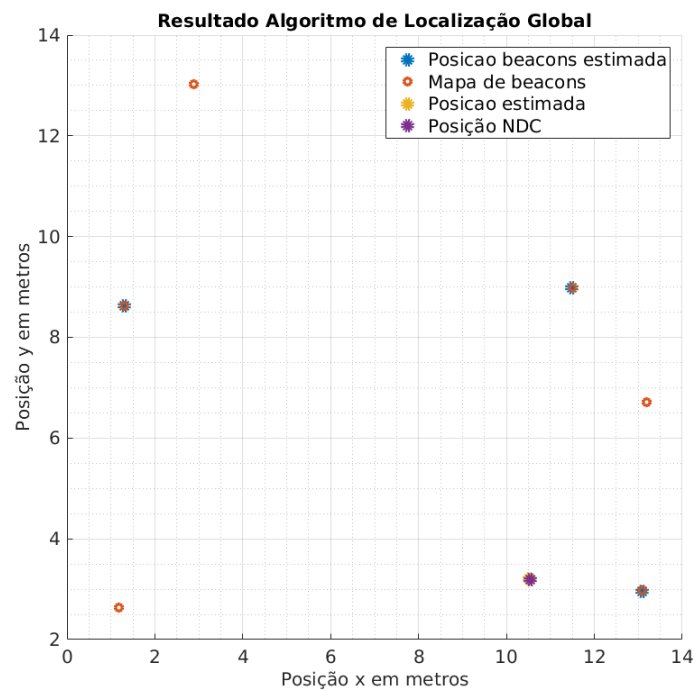


Figura 5.42: Resultado do algoritmo de localização global

	Pose NDC	Pose Algoritmo de Localização Global
x	10.54 m	10.51 m
y	3,19 m	3.20 m
theta	-0.89°	-0.96 °

Tabela 5.12: Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com três refletores detetados sem *outliers*

A diferença absoluta de posição e a diferença de ângulo da posição calculada pelo algoritmo em relação à pose dada pelo sistema da NDC foram as seguintes:

- *Diferença absoluta de posição: 0.032 m;*
- *Diferença de ângulo: -0.08 °.*

Para testar a robustez e a filtragem de *outliers* do algoritmo foram adicionados pontos aleatórios ao conjunto de três refletores observados no teste anterior para simular possíveis *outliers*.

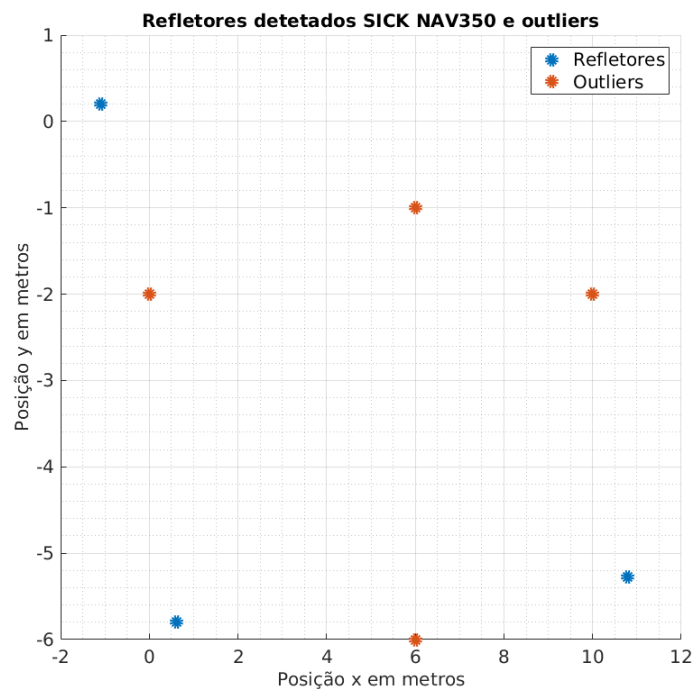


Figura 5.43: Conjunto de refletores detetados pelo laser *SICK NAV350* com *outliers*

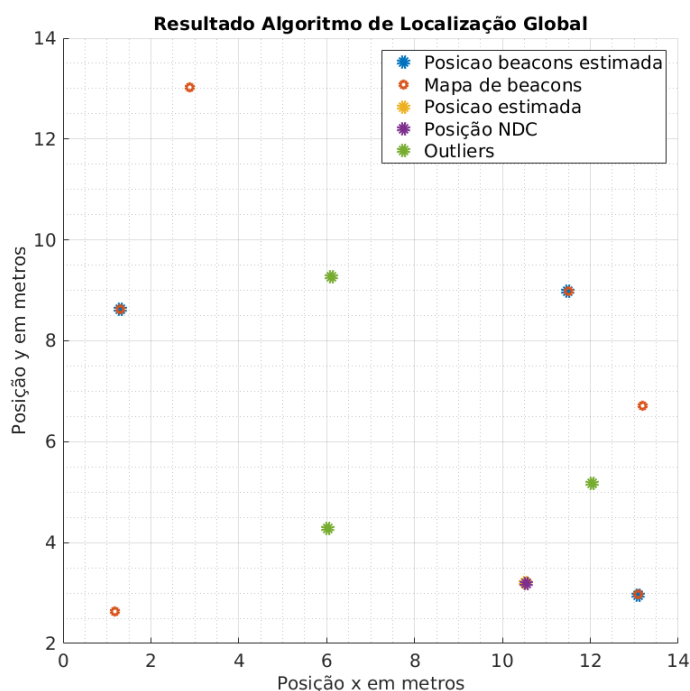


Figura 5.44: Resultado do algoritmo de localização global

Como se pode verificar na lista a seguir a pose calculada pelo algoritmo de localização global foi igual à anterior. Isto deveu-se ao facto de os *outliers* terem sido filtrados e o calculo da transformação entre os referencias ter sido efetuado apenas com as posições dos refletores reais, ou seja, as posições usadas no teste anterior sem *outliers*.

	Pose NDC	Pose Algoritmo de Localização Global
x	10.54 m	10.51 m
y	3,19 m	3.20 m
theta	-0.89°	-0.96 °

Tabela 5.13: Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com três refletores detetados com *outliers*

E conseqüentemente a diferença entre poses foi:

- *Diferença absoluta de posição: 0.032 m;*
- *Diferença de ângulo: -0.08 °.*

Além destes testes com o robô parado na sua posição inicial e a detetar apenas três refletores, o algoritmo foi testado com o robô noutras poses e a detetar um número superior de refletores.

No teste a seguir o AGV estava a detetar todos os refletores do ambiente de testes e não foram introduzidos *outliers*.

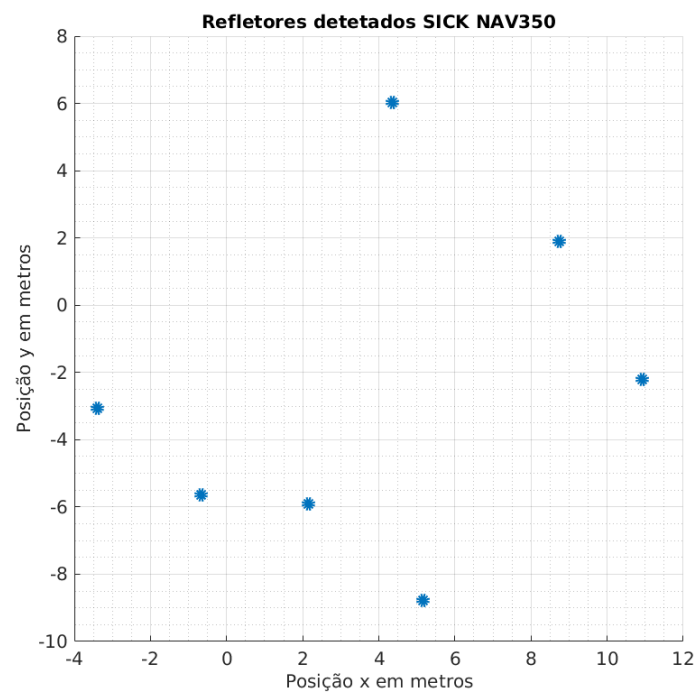
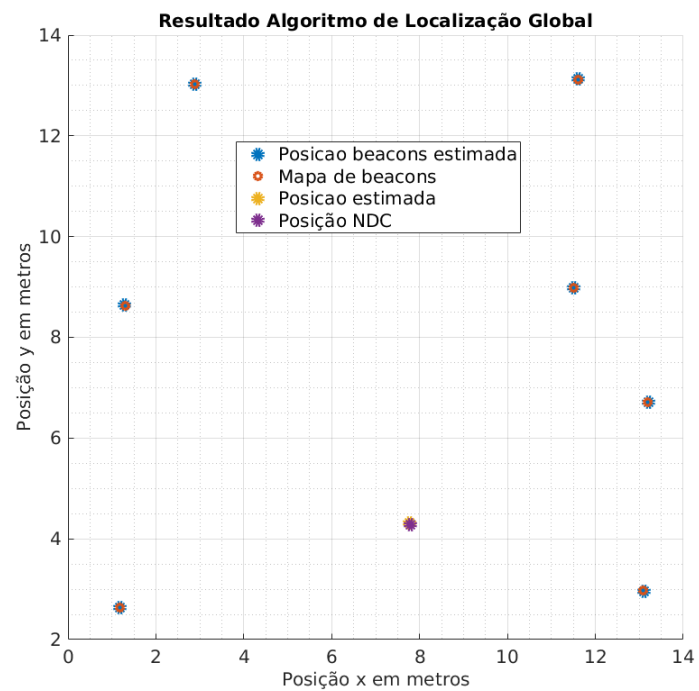
Figura 5.45: Conjunto de refletores detetados pelo laser *SICK NAV350*

Figura 5.46: Resultado do algoritmo de localização global

	Pose NDC	Pose Algoritmo de Localização Global
x	7.78 m	7.76 m
y	4.28 m	4.31 m
theta	-46.06°	-47.90 °

Tabela 5.14: Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com sete refletores detetados sem *outliers*

- *Diferença absoluta de posição: 0.041 m;*
- *Diferença de ângulo: - 1.85 °.*

Nestas condições e como se pode verificar na lista a seguir o erro absoluto de posição foi idêntico ao das experiências anteriores. É importante referir que a posição onde o AGV se encontra coincide com uma das curvas que o mesmo executa no percurso. Como se verificou nas comparações entre algoritmos, nestes locais registam-se os maiores desvios em relação à médio. Pode ser devido a isto que apesar de se estar a utilizar mais refletores no algoritmo de localização global, o mesmo retorne uma diferença maior em relação à pose do sistema da NDC, comparativamente com a experiência onde se utilizam apenas três refletores.

A seguir será feito o teste onde foram utilizados os refletores detetados na experiência anterior e adicionados *outliers* ao conjunto desses refletores. Ao conjunto de sete refletores detetados foram adicionados oito *outliers*. Com isto existem mais *outliers* do que refletores.

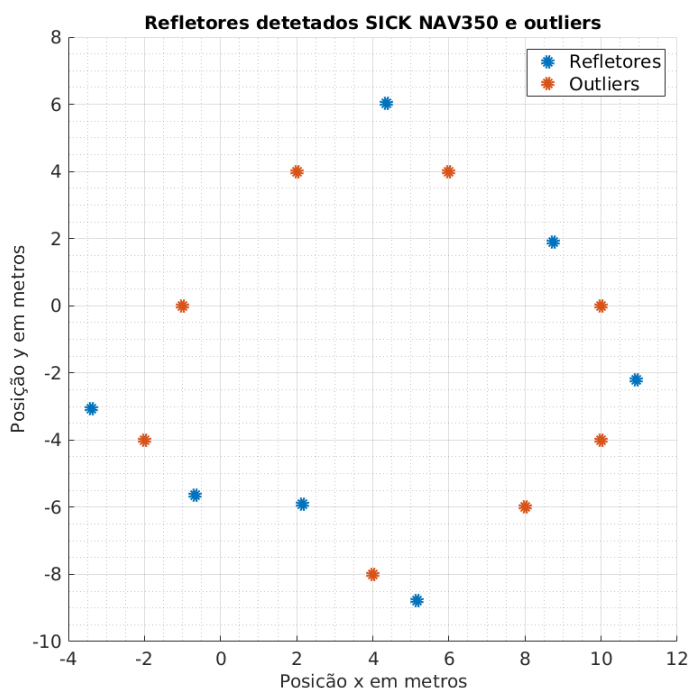


Figura 5.47: Conjunto de refletores detetados pelo laser *SICK NAV350*, com *outliers* adicionados

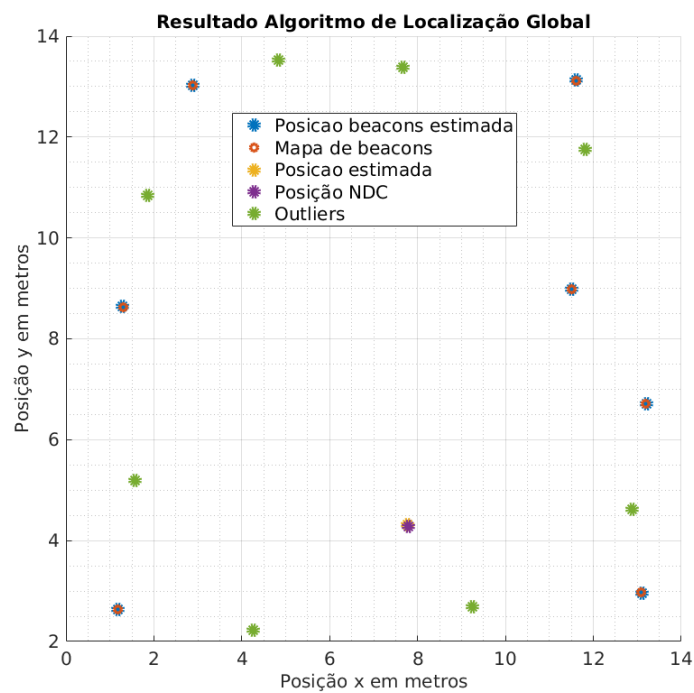


Figura 5.48: Resultado do algoritmo de localização global

Como se verifica na figura 5.48 o algoritmo convergiu para a solução correta apesar da quantidade de *outliers* adicionados às observações. Como era de esperar, e como aconteceu no teste anterior com três refletores e três *outliers* a pose calculada pelo algoritmo de localização global no teste com *outliers* foi idêntica à pose calculada pelo mesmo algoritmo mas sem *outliers*, uma vez que os *outliers* foram devidamente filtrados e foram utilizados apenas os verdadeiros refletores para calcular a pose do AGV.

	Pose NDC	Pose Algoritmo de Localização Global
x	7.78 m	7.76 m
y	4.28 m	4.31 m
theta	-46.06°	-47.90 °

Tabela 5.15: Pose calculada pelo sistema da NDC e pose calculada pelo algoritmo de localização global para a experiência com sete refletores detetados com *outliers*

- *Diferença absoluta de posição: 0.041 m;*
- *Diferença de ângulo: - 1.85 °.*

5.11 Ground Truth

Nas comparações anteriores foi possível perceber principalmente a diferença de resultados entre dois sistemas comerciais de localização, o sistema da NDC e a localização por parte do laser *SICK NAV350* relativamente a dois sistemas ainda pouco utilizados comercialmente, são eles o *Extended Kalman Filter* e o *Perfect Match*. No entanto não é ainda possível perceber qual deles se aproxima mais da posição real do robô.

O desafio neste ponto prende-se por determinar a posição real do robô x_i, y_i no referencial global. Uma vez que o sistema de visão que possibilita isso ainda não está operacional, optou-se por uma estratégia que analisa o deslocamento do AGV apenas sobre um eixo.

Para tal, colocou-se o AGV a executar uma trajetória retilínea entre os pontos 5 e 4 da figura 5.27 e mediu-se com um medidor de distâncias a laser o deslocamento do robô entre os dois pontos. Além disso, foram registadas as posições calculadas por todos os sistemas de localização ao longo dessa trajetória para calcularmos o deslocamento a partir de cada uma e calcular com o deslocamento calculado a partir das medidas do laser de medição.

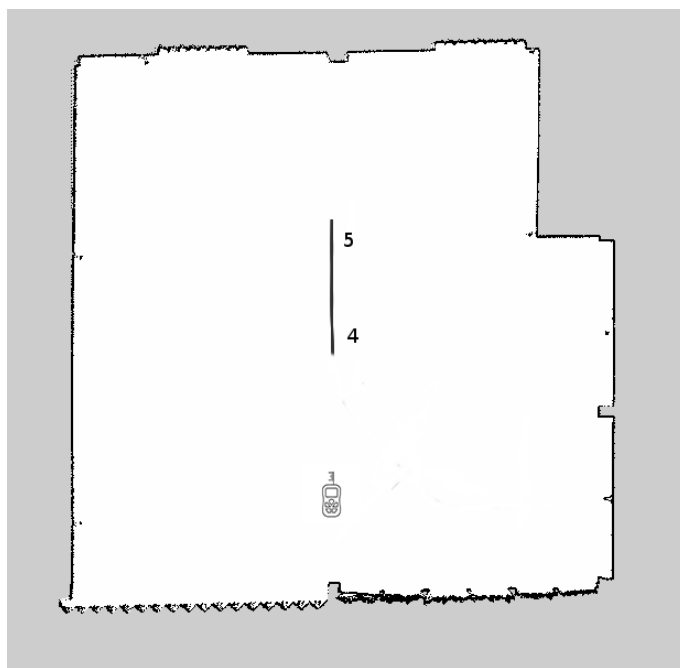


Figura 5.49: Montagem do teste para comparação entre o deslocamento registado pelo laser de medição com o deslocamento registado pelos sistemas de localização da NDC, *SICK NAV350*, *Extended Kalman Filter* e *Perfect Match*

Como mostra a figura 5.49, colocou-se um laser de medição numa posição colinear com a trajetória que irá ser executada pelo AGV e de forma a apontar sempre para o mesmo ponto no AGV. Em primeiro lugar colocou-se o robô na posição 5, mediu-se a distância registada pelo laser de medição e ligaram-se os sistemas de localização todos em simultâneo. Após isso o robô

executou uma trajetória retilínea até ao ponto 4 onde de novo se registou a medida dada pelo laser de medição.

Esta experiência foi executada três vezes seguidas e os valores de deslocamento obtidos foram os representados na tabela 5.16.

	Ensaio 1	Ensaio 2	Ensaio 3
Deslocamento NDC	2.2570 m	2.3610 m	2.4850 m
Deslocamento NAV350	2.2646 m	2.3682 m	2.4859 m
Deslocamento EKF	2.2677 m	2.3771 m	2.4991 m
Deslocamento Perfect Match	2.2689 m	2.3720 m	2.4914 m
Deslocamento medido no laser	2.2720 m	2.3780 m	2.5110 m

Tabela 5.16: Deslocamento registado pelo laser de medição com o deslocamento registado pelos sistemas de localização da NDC, *SICK NAV350*, *Extended Kalman Filter* e *Perfect Match*

	Ensaio 1	Ensaio 2	Ensaio 3
Erro NDC	0.0150 m	0.0170 m	0.0260 m
Erro NAV350	0.0074 m	0.0098 m	0.0251 m
Erro EKF	0.0043 m	0.0009 m	0.0119 m
Erro Perfect Match	0.0031 m	0.0060 m	0.0196 m

Tabela 5.17: Erro entre o deslocamento registado pelo laser de medição com os deslocamentos registados pelos sistemas de localização da NDC, *SICK NAV350*, *Extended Kalman Filter* e *Perfect Match*

Como se verifica na tabela 5.17 os erros mais baixos foram obtidos não pelos sistemas comerciais da NDC e do laser *SICK NAV350* mas sim pelos sistemas desenvolvidos pelo INESC-TEC *Extended Kalman Filter* e *Perfect Match*. Tal padrão aconteceu em todos os ensaios.

No que diz respeito aos sistemas comerciais, o sistema de localização do laser *SICK NAV350* registou em todos os ensaios, erros inferiores ao sistema da NDC.

Capítulo 6

Conclusão

Esta dissertação focou-se na comparação de quatro sistemas de localização, três com base em refletores, são eles o sistema da NDC8 Kollmorgen, o algoritmo de localização do próprio laser *SICK NAV350* e o *Extended Kalman Filter* e um com base em contornos, o *Perfect Match*. Destes quatro sistemas, dois deles são comerciais, que é o caso do sistema da NDC8 Kollmorgen e do algoritmo de localização do próprio laser *SICK NAV350* e dois foram desenvolvidos no INESC-TEC, que é o caso do *Extended Kalman Filter* e o *Perfect Match*. Dentro disso, foram comparados os métodos de localização de refletores do sistema comercial do laser *SICK NAV350*, com o desenvolvido no INESC-TEC, o "*Laser Scan to Beacons*". No entanto, estas comparações dão-nos valores de diferenças relativas entre os sistemas, por isso fizeram-se ensaios comparativos dos sistemas de localização com um *Ground Truth*. Por fim foi apresentado um algoritmo de localização global para o calculo da posição e orientação do robô no mundo.

Dos testes com os sistemas de localização, NDC8 Kollmorgen, *SICK NAV350*, *Extended Kalman Filter* e o *Perfect Match* todos apresentaram resultados satisfatórios numa análise superficial à posição e orientação que todos iam apresentando durante os ensaios realizados. Numa análise mais detalhada o sistema comercial da NDC8 Kollmorgen apresentou a posição e orientação mais suave de todos os sistemas, seguindo-se o sistema desenvolvido pelo INESC-TEC *Extended Kalman Filter*, depois o sistema comercial *SICK NAV350* e por último o algoritmo de localização com base em contornos desenvolvido pelo INESC-TEC *Perfect Match*.

Numa análise relativa com os sistemas comerciais, NDC8 Kollmorgen e *SICK NAV350* aquele que apresenta uma maior diferença de posição e orientação é o *Perfect Match*. Tal resultado era esperado pois ao contrário destes três sistemas, o algoritmo *Perfect Match* não utiliza refletores para calcular a posição e orientação do robô mas sim os contornos do meio de navegação. Como foi dito na revisão bibliográfica, os sistemas que usam refletores são os que têm apresentado melhores resultados relativamente a outras soluções existentes. No entanto a diferença para os restantes algoritmos não foi muito elevada. Isto é de realçar pois estas diferenças relativamente baixas foram relativas a sistemas comerciais já bastante utilizados.

Foi comparada também a localização de refletores por parte do sistema comercial *SICK NAV350* com o sistema desenvolvido pelo *Laser Scan to Beacons*, sendo que o sistema comercial apre-

sentou melhores resultados que o sistema do INESC-TEC. No entanto, o *Extended Kalman Filter* apresentou resultados muito semelhantes, quer a usar a detecção de refletores do laser *SICK NAV350*, quer a partir do *Laser Scan to Beacons*.

Quando se comparou os vários sistemas de localização com o *Ground Truth*, o sistema que apresentou menor erro foi o *Perfect Match*, seguindo-se o *Extended Kalman Filter*, depois o *SICK NAV350* e por último o *NDC*.

Por fim o algoritmo de localização global proposto e testado com dados do laser *SICK NAV350* apresentou resultados bastante satisfatórios quer a detetar o número máximo de refletores existente no meio, com ou sem *outliers* quer a detetar o número mínimo de refletores para que o algoritmo possa calcular uma posição e orientação, ou seja três, com ou sem refletores.

Capítulo 7

Trabalho futuro

Já numa fase final da elaboração da dissertação surgiu a ideia de melhorar ainda mais a robustez a *outliers* do algoritmo de localização global proposto. A ideia seria identificar as distâncias entre refletores detetados que fossem maiores que a distância máxima entre os refletores do mapa e as distâncias menores que a distância mínima entre os refletores do mapa, eliminando essas distâncias.

O algoritmo de localização global aqui proposto foi implementado em *Matlab* e testado com dados reais obtidos em vários testes com o AGV. No entanto, para o uso deste algoritmo nos projetos desenvolvidos pelo INESC-TEC é de especial interesse a implementação deste algoritmo em ROS. Após a implementação do algoritmo de localização global em ROS é possível fazer a sua integração noutros algoritmos de localização, como é o caso do *Extended Kalman Filter*. Este poderia ser integrado no EKF para funcionar da forma apresentada na seguinte máquina de estados da figura 7.1.

Como foi dito anteriormente, os dois lasers utilizados não estão perfeitamente alinhados. Este facto introduz uma pequena diferença entre as medições obtidas pelo laser da NDC8 e as medições do laser *SICK NAV350*. Para que se possam obter comparações ainda mais precisas é de importante relevância a calibração da posição dos lasers de navegação.

A estratégia de *Ground Truth* utilizada apenas permite validar o deslocamento linear num dos eixos. Para que seja possível fazer uma análise mais detalhada acerca dos erros de cada algoritmo em relação à posição real do robô será necessária a utilização de outra estratégia que permita saber a posição e orientação do AGV ao longo da trajetória.

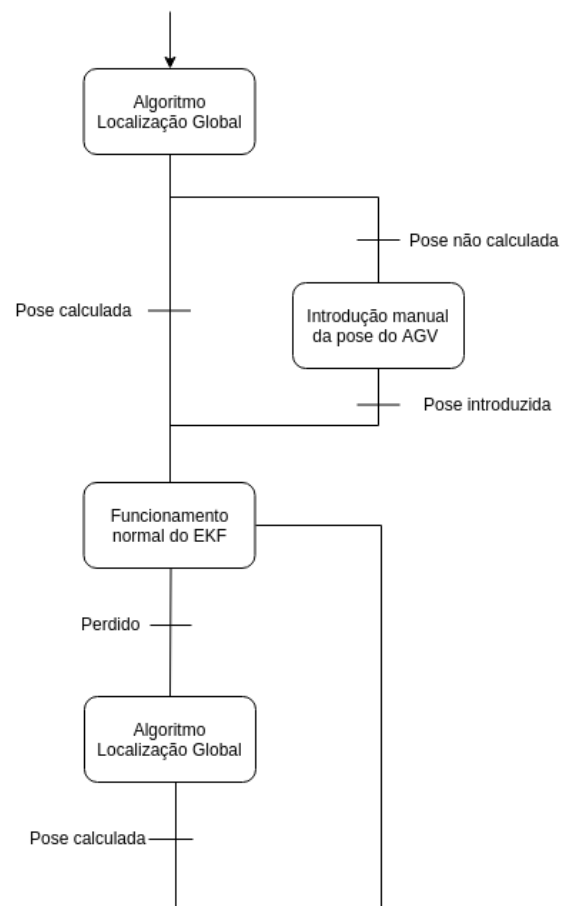


Figura 7.1: Máquina de estados para integração do algoritmo de localização global com o *Extended Kalman Filter*

Referências

- P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. ISSN 0162-8828. doi: 10.1109/34.121791.
- João Sena Esteves. Metodologia de autolocalização absoluta em ambientes quase-estruturados. 2005.
- Guangjie Han, Huihui Xu, Trung Q. Duong, Jinfang Jiang, and Takahiro Hara. Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 52(4):2419–2436, Apr 2013. ISSN 1572-9451. doi: 10.1007/s11235-011-9564-7. URL <https://doi.org/10.1007/s11235-011-9564-7>.
- C. Hekimian-Williams, B. Grant, Xiuwen Liu, Zhenghao Zhang, and P. Kumar. Accurate localization of rfid tags using phase difference. In *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, pages 89–96, April 2010. doi: 10.1109/RFID.2010.5467268. URL <http://ieeexplore.ieee.org/abstract/document/5467268/>.
- Martin Lauer, Sascha Lange, and Martin Riedmiller. Calculating the perfect match: An efficient and accurate approach for robot self-localization. In Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, pages 142–153, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-35438-3.
- David Lima et al. Localização absoluta de robôs móveis em ambientes industriais. 2010.
- V. Pierlot and M. Van Droogenbroeck. Beams: A beacon-based angle measurement sensor for mobile robot positioning. *IEEE Transactions on Robotics*, 30(3):533–549, June 2014. ISSN 1552-3098. doi: 10.1109/TRO.2013.2293834.
- Christoph Reinke and Patric Beinschob. Strategies for contour-based self-localization in large-scale modern warehouses. In *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*, pages 223–227. IEEE, 2013.
- D. Ronzoni, R. Olmi, C. Secchi, and C. Fantuzzi. Agv global localization using indistinguishable artificial landmarks. In *2011 IEEE International Conference on Robotics and Automation*, pages 287–292, May 2011a. doi: 10.1109/ICRA.2011.5979759.
- Davide Ronzoni, Roberto Olmi, Cristian Secchi, and Cesare Fantuzzi. Agv global localization using indistinguishable artificial landmarks. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 287–292. IEEE, 2011b.

- L. Schulze and A. Wullner. The approach of automated guided vehicle systems. In *2006 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 522–527, June 2006. doi: 10.1109/SOLI.2006.328941. URL <http://ieeexplore.ieee.org/document/4125635/>.
- Lothar Schulze, Sebastian Behling, and Stefan Buhrs. Automated guided vehicle systems: a driver for increased business performance. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 2, 2008.
- Heber Sobreira, Luís Rocha, Carlos Costa, José Lima, Paulo Costa, and A Moreira. 2d cloud template matching - a comparison between iterative closest point and perfect match. 05 2016.
- Héber Miguel Plácido Sobreira. Fiabilidade e robustez da localização de robôs móveis. 2017. URL <http://hdl.handle.net/10216/105452>.
- Olga Sorkine. Least-squares rigid motion using svd. *Technical notes*, 120(3):52, 2009.
- Sebastian Thrun, Michael Beetz, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Haehnel, Chuck Rosenberg, Nicholas Roy, et al. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research*, 19(11):972–999, 2000.
- Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.
- Ke Wang, Ampalavanapillai Nirmalathas, Christina Lim, Kamal Alameh, Hongtao Li, and Efstratios Skafidas. Indoor infrared optical wireless localization system with background light power estimation capability. *Opt. Express*, 25(19):22923–22931, Sep 2017. doi: 10.1364/OE.25.022923. URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-25-19-22923>.
- H. Zhang, L. Zhang, and J. Dai. Landmark-based localization for indoor mobile robots with stereo vision. In *2012 Second International Conference on Intelligent System Design and Engineering Application*, pages 700–702, Jan 2012. doi: 10.1109/ISdea.2012.640.
- Junyi Zhou and Jing Shi. Rfid localization algorithms and applications—a review. *Journal of Intelligent Manufacturing*, 20(6):695, Aug 2008. ISSN 1572-8145. doi: 10.1007/s10845-008-0158-5. URL <https://doi.org/10.1007/s10845-008-0158-5>.